Vehicle Routing and Job Shop Scheduling: What's the difference?*

J. Christopher Beck
Cork Constraint Computation Center
University College Cork, Ireland
c.beck@4c.ucc.ie

Patrick Prosser and **Evgeny Selensky**
Department of Computing Science,
University of Glasgow, Scotland
{pat, evgeny}@dcs.gla.ac.uk

Abstract

Despite a number of similarities, vehicle routing problems and scheduling problems are typically solved with different techniques. In this paper, we undertake a systematic study of problem characteristics that differ between vehicle routing and scheduling problems in order to identify those that are important for the performance of typical vehicle routing and scheduling techniques. In particular, we find that the addition of temporal constraints among visits or the addition of tight vehicle specialization constraints significantly improves the performance of scheduling techniques relative to vehicle routing techniques.

Introduction

It is a long standing belief in AI that finding the right problem representation is a key component of solving a problem (Fink 1995; 1998; Smirnov & Veloso 1996). Given the growth in knowledge of combinatorial problems, a common heuristic is to model a new problem as a known combinatorial problem for which there exist strong solution methods (Walsh 2000). For example, if a problem can be modeled as a graph colouring problem or even better a linear program, there are existing techniques that are very likely to solve the problem adequately.

The reality of real world problems is that it is likely that in formulating a problem as an instance of an existing class, some compromises must be made. There will be aspects of the real problem that do not fit easily into a "pure" version of a known combinatorial optimization problem. The decision must then be made to ignore the problem components that cannot be modeled in a pure way or to add "impure" or customized model components. The drawback of the former approach is that it is not clear that solutions to the problem model are solutions to the real problem. The latter approach also presents challenges, due to the fact that we do not yet understand combinatorial problem solving well enough to be able to predict the effect of adding a customized model component. It can be that what looks like a minor addition to the model significantly changes the performance of standard problem solving techniques.

A case in point are real world scheduling problems. The AI and Operations Research literature has delivered strong problem solving techniques for a variety of pure problem definitions (e.g., job shop scheduling problem (JSP), resource constraint project scheduling problem (RCPSP)). It is rare for a pure JSP to be found in the wild. There are typically additional constraints to deal with like transition times between tasks, multiple (contradictory) optimization criteria, and resource alternatives. In fact, when faced with a real problem it is sometimes difficult to decide which type of pure problem model to start with: though it may look like a scheduling problem perhaps some characteristics mean that vehicle routing technology may be a better choice. For a complicated problem with some scheduling and some vehicle routing characteristics, we do not yet understand the problem models and problem solving techniques well enough to know which pure model is a better fit.

This paper directly addresses the relationship between problem characteristics and search performance. In previous work (Selensky 2001), it was shown that scheduling technology performs poorly on reformulated vehicle routing problems while vehicle routing technology performs poorly on reformulated scheduling problems. Here, we investigate the reasons behind these results. We perform an asymmetric study by systematically mutating pure vehicle routing problems (VRP) such that they become more like scheduling problems. We then solve these problems with routing and with scheduling technologies, to discover what problem characteristics influence the problem solving technologies.[1]

This paper is organized as follows. We start by defining vehicle routing problems and scheduling problems, and show how we can reformulate VRPs as scheduling problems, and vice versa. We then identify five characteristics that we believe differentiate routing problems from scheduling problems. We describe the design of our experiments, to investigate the behaviour of these problem characteristics on the relative performance of routing and scheduling technology. The results of these experiments are reported, and we close with a conclusion and plan of future work.

*This work was supported by EPSRC research grant GR/M90641, Science Foundation Ireland under Grant 00/PI.1/C075, and ILOG SA.

[1] In a future study we will start with jobshop scheduling problems and mutate these such that they become more VRP-like.

Problem Definitions: VRP & JSP

In the capacitated vehicle routing problems with time windows (VRPTW), m identical vehicles initially located at a depot are to deliver discrete quantities of goods to n customers. Each customer has a demand for goods and each vehicle has a capacity. A vehicle can make only one tour starting at the depot, visiting a subset of customers and returning to the depot. Time windows define an interval for each customer within which the visit must be made. A solution is a set of tours for a subset of vehicles such that all customers are served only once and time window and capacity constraints are respected. The objective is to minimize distance traveled, and sometimes additionally to reduce the number of vehicles used. The problem is NP-hard (Garey & Johnson 1979).

An $n \times m$ job shop scheduling problem (JSP) consists of n jobs and m resources. Each job is a set of m completely ordered activities, where each activity has a duration for which it must execute and a resource which it must execute on. The total ordering defines a set of precedence constraints, meaning that no activity can begin execution until the activity that immediately precedes it in the complete ordering has finished execution. Each of the m activities in a single job requires exclusive use of one of the m resources defined in the problem. No activities that require the same resource can overlap in their execution and once an activity is started it must be executed for its entire duration (i.e. no pre-emption is allowed). The job shop scheduling decision problem is to decide if all activities can be scheduled, given for each job a release date of 0 and a due date of the desired makespan D, while respecting the resource and precedence constraints. The job shop scheduling decision problem is also NP-complete (Garey & Johnson 1979).

While not part of the basic JSP definition, transition times and resource alternatives have been reported in the scheduling literature (Focacci, Laborie, & Nuijten 2000). In the case of a transition time, there is a temporal constraint specifying a minimum time that must expire between pairs of activities executed on the same resource. When alternative resources are represented, each activity has a set of resources that it can be performed on (and this extension of the JSP is known as the flexible job shop problem).

Mutual Reformulations

In (Beck, Prosser, & Selensky 2002) two reformulations are presented: a reformulation of VRP as a flexible shop problem with transition time, and symmetrically a reformulation of JSP as a VRP.

We reformulate the VRP into a scheduling problem as follows. Each vehicle is represented as a resource, and each customer visit as an activity. The distance between a pair of visits corresponds to a transition time between respective activities. Each activity can be performed on any resource, and is constrained to start execution within the time window defined in the original VRP. Each activity has a demand for a secondary resource, corresponding to a visit's demand within a vehicle. For each resource R there are two special activities $Start_R$ and End_R. Activities $Start_R$ and End_R must be performed on resource R. $Start_R$ must be the first activity performed on R and End_R the last. The transition time between $Start_R$ and any other activity A_i corresponds to the distance between the depot and the i^{th} visit. Similarly the transition time between End_R and A_i corresponds to the distance between the depot and the i^{th} visit. The processing time of $Start_R$ and End_R is zero. We associate a consumable secondary resource with every (primary) resource to model the capacity of vehicles. Consequently a sequence of activities on a resource corresponds to a vehicle's tour in the VRP. In the resultant scheduling problem each job consists of only one activity, each activity can be performed on any resource, and there are transition times between each pair of activities. The problem is then to minimize the sum of transition times on all machines and maybe also to minimize the number of resources used.

We reformulate a scheduling problem into a VRP as follows. We have for each resource, a vehicle, and for each activity, a customer visit. The visits have a duration the same as that of the corresponding activities. Each visit can be made only by the vehicles corresponding to the set of resources for the activity. Any ordering between activities in a job results in precedence constraints between visits. Transition times between activities correspond to travel distances between visits. The deadline D imposes time windows on visits. Assuming we have m resources, and therefore m vehicles, we have $2m$ dummy visits corresponding to the departing and returning visits to the depot. A vehicle's tour corresponds to a schedule on a resource. For the $n \times m$ JSP we have a VRP with $m(n+2)$ visits and m vehicles. Each visit can be performed only by one vehicle. Since there are no transition times in the pure JSP, there are no travel distances between visits, but visits have durations corresponding to those of the activities. There are precedence constraints between those visits corresponding to activities in a job. The decision problem is then to find an ordering of visits on vehicles that respects the precedence constraints and time windows.

Problem Characteristics

The performance difference between VRP and scheduling techniques must be due to the characteristics on which the problems differ. In this section, we identify five characteristics that we believe are sufficient to explain the performance differences between the VRP and scheduling techniques.

Alternative Resources Perhaps the most obvious difference between standard VRP and scheduling problems is the number of alternative resources that may be used for each operation. In vehicle routing there are typically many vehicles that can be used to perform a visit. For example, the standard Solomon benchmarks (Solomon 1987) have twenty five identical vehicles. In real problems, it is often the case that vehicles are partitioned into classes and each visit can only be performed by vehicles in one class. This technique can be used to model different forms of transportation (e.g., truck vs. plane) or different characteristics of the vehicle (e.g., a refrigerated vehicle must be used to deliver ice cream). However, even with such partitions there are typically many vehicles that can perform each visit. This characteristic of pure VRP problems is a reflection of the real world problems in which delivery vehicles are not usually specialized in the delivery of particular products: for most visits any vehicle will suffice.

In contrast, scheduling problems tend to have very few alternative resources. As noted, in the pure JSP there are no resource alternatives. This has been reflected in the scheduling literature. For example, Focacci et al. (Focacci, Laborie, & Nuijten 2000) experiment on problems with up to three alternatives while Davenport & Beck (Davenport & Beck 1999) use problem instances with at most eight alternatives. As with VRPs, the reasons for this restriction in scheduling problems stem from the real world situation. In a factory, machines are often specialized for a particular task (or a particular set of related tasks). In an assembly line for example, alternatives may arise based on selecting which line an order should be produced on, but within a line there are typically few resource alternatives. This characterization of scheduling problems with few resource alternatives is of course a generalization. There are scheduling problems with significant resource alternatives. For example, with modern computer controlled machines, one machine may be reconfigurable to perform many different activities. Indeed, the existence of such scheduling problems is one of the motivations for this study since it is not clear in just such a case as to whether traditional scheduling or VRP solution techniques should be applied.

Temporal Constraints In pure VRPs each visit is independent, i.e. there are no constraints requiring a visit to have some temporal relation with other visits. In the pickup-and-delivery variant (Toth & Vigo 2002) pairs of activities are related such that they both must be on the same vehicle and must occur in a prescribed order. This characteristic again arises from real world problems where visits tend to be truly independent. However, temporal relations between visits do arise. In a study of workforce management for British Telecom (Brind, Muller, & Prosser 1995) service engineers are routed to customers, to install and repair equipment. In this domain, temporal constraints are ubiquitous: certain installation tasks must be done in different locations in sequence, and sometimes simultaneously (such as end to end tests).

In contrast, scheduling problems typically have long chains or complex directed acyclic graphs of temporal relationships among activities. As a consequence temporal reasoning has been widely explored in the literature (Cesta, Oddi, & Smith 2000) and is an extremely important component of many scheduling algorithms (Laborie 2001). This prominence arises from the need to execute a set of temporally related steps in order to successfully generate a product. For example, in the chemical or pharmaceutical industries the timing and sequencing of reactions is critical.

The Ratio of Operation Duration to Transition Time In pure VRPs a visit has no duration. In contrast, in pure JSPs the transition time between operations is zero. The modeling abstraction for scheduling is obviously opposite to VRP: the transition time between activities on the same resource is so small that it can be ignored. These two problem models are extremes. In both the literature and in the real world there are VRPs with visit duration and scheduling problems with transition times and costs. However, even then the ratio of operation duration to transition time tends to be different. In VRPs the ratio is very small while in scheduling it is often very large. This difference is reflected in solution techniques. Many VRP techniques (e.g., the savings heuristic (Clarke & Wright 1964)) look exclusively at transition times when searching, while many of the strong constraint propagation (Nuijten 1994; Le Pape 1994) and heuristic search techniques (Smith & Cheng 1993; Beck & Fox 2000) in scheduling ignore transition time.

Optimization Criterion In a VRP the standard optimization criterion is to minimize the total distance traveled by each vehicle.[2] In scheduling, a common criterion is the minimization of makespan: the time between the start of the first operation and the end of the last operation. While many criteria have been studied (e.g., tardiness (Baptiste, Le Pape, & Nuijten 2001), earliness and tardiness (Beck & Refalo 2002), flow time, transition times (Focacci, Laborie, & Nuijten 2000)) makespan has received a great deal of attention in the literature even though its relevance has been questioned in practical applications (Fox 1983).

Makespan arises as a criterion due to physical relationship of the machines in a factory and the associated overhead cost. Often a factory needs to be running from the beginning of the first task to the end of the last, regardless of the number of machines that are actually executing operations. If the overhead cost is significant, the goal of minimizing makespan is a reasonable representation of the goal of cost minimization. In contrast, the vehicles in a VRP have no such physical relationship: they are typically independent and the overhead of the depot is much lower than the cost of travel. Therefore, the overall cost can be calculated on a per vehicle basis rather than based on the overhead of the factory.

Temporal Slack Although it seems difficult to make specific predictions as to how temporal slack, i.e., the difference between the time window for an operation and the operation's duration, can affect the performance of the search techniques in both routing and scheduling, we think it worthwhile to experiment with it as well. Slack can be important while solving both routing or scheduling problems, e.g., there are state-of-the-art global constraint propagation algorithms and efficient search heuristics based on temporal slack (Smith & Cheng 1993).

Summary We have identified five problem characteristics that we consider sufficient to explain the performance differ-

[2]It is also common to minimize the number of vehicles used while minimizing the travel distance. For this paper, we focus on minimization of travel time.

ence between VRP and scheduling technology. These characteristics are:

1. Alternative resources – We expect few resource alternatives to favour scheduling techniques while many should improve the performance of VRP techniques.
2. Temporal constraints – Many and more complex temporal constraints are characteristic of scheduling problems therefore we predict that scheduling technology should improve on problems where there are complex temporal constraints.
3. Operation duration vs. transition time – Transition time can be vanishingly small in scheduling problems while task duration can be similarly insignificant in VRPs. Therefore, the smaller the ratio of operation duration to transition time, the better the VRP techniques should perform relative to the scheduling techniques.
4. Optimization criterion – When the optimization criterion is the minimization of makespan as opposed to the minimization of total transition time, we predict that the performance of the scheduling techniques should be favoured.
5. Temporal Slack – slack tends to be large in VRPs, if only because the duration of visits are small. Scheduling technology has developed special purpose propagators and heuristics to cope with slack. How will variations in slack effect technology performance?

These characteristics may be sufficient to explain the performance differences between VRP and scheduling techniques, because they are sufficient to transform one problem model to the other. Starting with a pure VRP, if we reduce the alternative resources to a singleton for each task, add chains of temporal constraints, reduce the transition time to zero while increasing task duration, and change the optimization criterion to makespan minimization, we have a pure JSP problem.

Experimental Design

Our goal is to empirically determine which of the above five problem characteristics actually have an effect on the relative performance of scheduling techniques compared to VRP techniques. In our study we take pure VRP instances and examine the effect of varying one problem characteristic at a time. We generate instances of VRPs. We then solve each problem twice: first we model the problem as a VRP and solve it using routing technology; second, we model the VRP as a scheduling problem (as described in the earlier section Mutual Reformulations) and solve it with scheduling technology. We then measure the relative performance of the technologies. For each problem characteristic, therefore, we begin with a set (or ensemble) of pure VRP instances. We then generate additional problem sets by varying the parameter in question. Intuitively, as the parameter value deviates more from the VRP, the ensemble should look increasingly like scheduling problems.

Problems and Their Parameters

The initial set of pure VRP instances contains 100 problems. The geographic coordinates of each visit are randomly drawn with replacement from a data file containing all postal codes of locations in the city of Glasgow, within a 5 km radius of the city centre. These postal codes can be directly translated to coordinates with an accuracy of ten meters. Because we are using city coordinates, all distances are Manhattan distance.

The other parameters for our problem generator are as follows:

n Number of customers. In all our problems $n = 100$.

m Number of vehicles in the fleet. In all problems $m = 25$.

p Vehicle specialization. The parameter p represents the proportion of the fleet that can perform each visit. For visit i, the number of possible vehicles m_i is calculated as:

$$m_i = \lceil pm \rceil \tag{1}$$

Small values of p correspond to few resource alternatives and therefore correspond more to scheduling problems. The default value is $p = 1.0$

pc The (integer) number of precedence constraints. This parameter specifies the number of pairs of visits whose sequence is constrained such that visit i must end before the start of visit j. Note that i and j are not necessarily on the same vehicle. Large values of pc correspond to many precedence constraints characteristic of scheduling problems. The default value is $pc = 0$, i.e. no precedence constraints.

ρ The ratio of operation duration to travel time. This parameter is used to set V, the speed of the vehicles.

$$V = \frac{\rho.\overline{d}}{\overline{\tau}} \tag{2}$$

where $\overline{\tau}$ is the average duration of visits, and $\overline{d}$ is the average distance between visits. Large ρ values are typical of scheduling problems while small ρ values correspond to VRPs. The default value is $\rho = 1.0$.

σ Normalized slack.

$$\sigma = \frac{le_i - \tau_i - es_i}{le_i - es_i}, \tag{3}$$

where τ_i is the duration of operation i, and le_i and es_i are the latest end and earliest start of operation i. It follows from this definition that $\sigma \in (0, 1)$ and that the larger σ becomes, the wider the time window is for the same τ_i. The default value is $\sigma = 0.9$

c The optimization criterion. There are only two possible values for c: 'minimize makespan' and 'minimize total travel time'. The default criterion is 'minimize total travel time'.

Pure VRP problems are generated with the default parameters, specified above. Problems are tested for solubility, and are rejected if not found to be soluble within 5 minutes of CPU time.

Solution Technology

In our experiments we use the ILOG optimization suite. The scheduling library (ILOG Scheduler 5.2) has global constraint propagation (Baptiste, Le Pape, & Nuijten 2001; Laborie 2001; Nuijten 1994) within constructive tree search as its core technology, while the vehicle routing library (ILOG Dispatcher 3.2) focuses on local search (DeBacker *et al.* 2000).

As part of the routing technique, we apply the guided local search (GLS) metaheuristic (Voudouris & Tsang 1998) with a penalty factor of 0.4. To construct a neighbourhood of a solution, we use the standard move operators (*TwoOpt*, *OrOpt*, *Relocate*, *Cross* and *Exchange*). Search starts from the first solution constructed by the savings heuristic. Local search then descends to a local optimum. The GLS metaheuristic is then applied to drive the search out of that local optima. The scheduling technique uses constructive depth first search with standard slack based heuristics (Smith & Cheng 1993), global constraint propagation, and edge finding (Baptiste, Le Pape, & Nuijten 2001).

Evaluation Criteria

Our primary evaluation criteria is λ, the ratio of the cost of the best solution found by the scheduling technique to that found by the VRP technique in a given CPU time limit. Specifically:

$$\lambda = \frac{C_{sched}}{C_{rout}}, \tag{4}$$

where C_{sched} and C_{rout} are the cost of the best solutions found by the scheduling and routing techniques in the given amount of CPU time. When λ is larger than 1, routing technology is outperforming scheduling technology, when λ is less than 1 scheduling technology dominates, and when $\lambda = 1$ the technologies perform equally.

The CPU time bound is chosen to be 10 minutes. Our preliminary experiments showed that there is no significant improvement in solutions for either solving technology after 10 minutes. Fig. 1 plots the mean value of λ against CPU time for different values of ρ, and shows a point of diminishing return at about 10 CPU minutes.

The Empirical Results

We now study the influence of the following: variation in the number of alternative resources, the number of precedence constraints, the influence of operation duration versus transition times, optimization criteria, and finally slack.

Alternative Resources (p)

To investigate the influence of the number of resource alternatives we create a series of problem sets by varying the vehicle specialization p, such that $p \in \{0.05, 0.1, 0.25, 0.5, 1.0\}$. With $m = 25$ this corresponds to VRPs with 1, 3, 6, 13, and 25 possible vehicles per visit respectively. All other parameters take their default values (i.e. $pc = 0$, $\rho = 1.0$, $\sigma = 0.9$, and c minimizing total travel time).

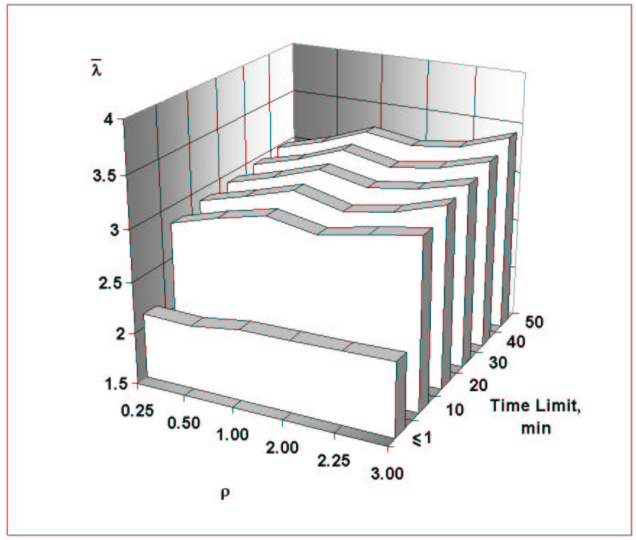

Figure 1: Influence of CPU time on λ for different values of ρ. Each value of λ is an average of 20 VRPs with $p = 1.0$ and $\sigma = 0.9$.

In Fig. 2 we present λ with respect to p, expressed as a percentage. Two contours are presented, one for λ for first solution found (λ_{first}) by both techniques, and λ for the best solution found (λ_{best}) by both techniques. We see that as specialization of the fleet increases (i.e. p decreasing) the performance of the routing technology degrades, and ultimately when we have one vehicle per visit the technologies perform approximately equally. Both contours exclude any instance that could not be solved by the routing technology in 10 minutes.

Fig. 3 shows how many instances per ensemble could not be solved by the routing technology. We observe an insolubility peak at $p = 0.1$. We conjecture that at $p = 0.1$ problems are critically constrained such that there are very few feasible vehicle assignments. Since savings is a single pass heuristic, it does no search for such assignments and therefore fails when they are difficult to construct. With $p < 0.1$, no vehicle choices are necessary since only one vehicle can perform each visit. At $p > 0.1$, many vehicles are possible, increasing the number of feasible vehicle assignments.

To test if the variation in λ is entirely due to the behavior of the initial construction using the savings heuristic, we allow the routing technique to start from a solution obtained by the scheduling technique. Fig. 4 plots a scatter of λ for best solutions, where both techniques have a common starting point, i.e. the first scheduler solution. In all cases the GLS component of the routing technology was able to improve the initial solution. However, we see a marked degradation in the overall performance of the VRP technology: the values of λ are significantly lower than in our previous experiment.

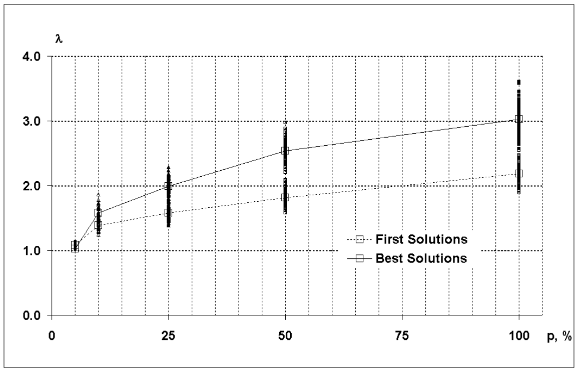

Figure 2: What happens when we specialize the fleet? p varies, $\rho = 1.0$, $\sigma = 0.9$.

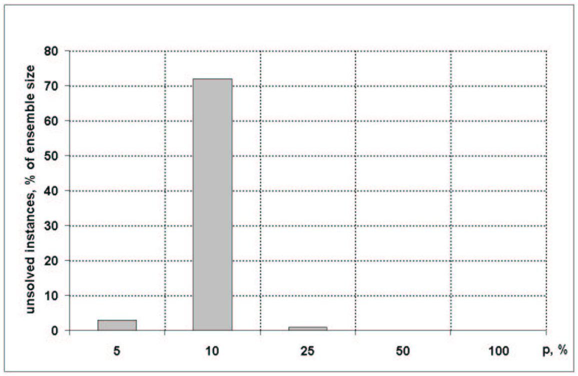

Figure 3: Percentage of instances unsolved by the routing technique *vs* p, $\rho = 1.0$, $\sigma = 0.9$.

Precedence Constraints (*pc*)

To study the influence of precedence constraints we vary pc, such that $pc \in \{0, 50, 450, 1200\}$. Each precedence constraint imposes an ordering for two given visits, without forcing these two visits to use the same vehicle. This corresponds to varying numbers and lengths of totally ordered precedence chains such that 5 chains each involving 5 visits are equivalent to a pc of 50, 10 chains with 10 visits each to a pc of 450, 4 chains with 25 visits each to a pc of 1200.

The routing technology turns out to be incapable of constructing first solutions when precedence constraints are introduced. In fact, even for a small number of precedences between visits ($pc = 50$), almost all instances in an ensemble could not be solved by this technique. Therefore we allow both technologies to share initial solutions, i.e. the VRP technology starts with the scheduling first solution, consequently all problems are again soluble. In Fig. 5 we see that as we increase the number of precedence constraints the scheduling technology marginally improves relative to the VRP technology (i.e. λ falls), although the VRP technology continues to dominate.

In Fig. 6 we plot the percentage reduction in cost of the final solution with respect to the initial scheduling solution for

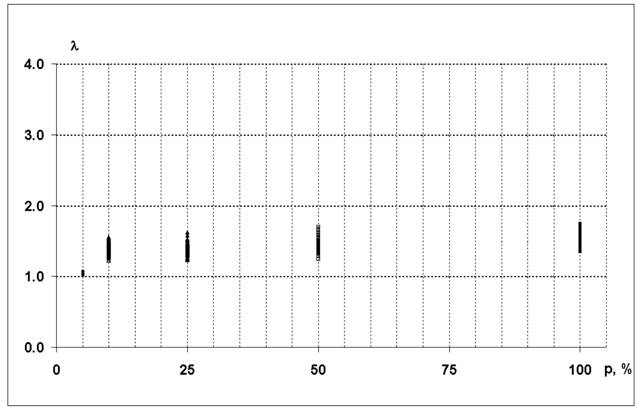

Figure 4: λ against p with a common initial (scheduling) solution. $\rho = 1.0$, $\sigma = 0.9$. All problems now have solutions.

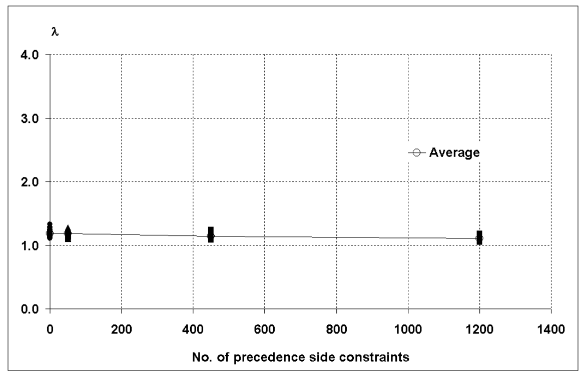

Figure 5: λ against pc the number of precedence constraints. $\rho = 1.0$, $\sigma = 0.9$, $p = 1.0$. First solutions were obtained by the scheduling technique and used by both the scheduling and routing techniques.

the VRP technology. We observe a decrease in improvement as we increase precedence constraints.

In conclusion, as we introduce precedence constraints the VRP technology fails to produce initial solutions, but the scheduling technology appears to be robust. If both techniques start with a scheduling solution, the VRP technology dominates the scheduling technology, but this becomes less significant as we increase the number of precedence constraints.

Operation Duration vs Transition Time (ρ)

VRP instances were generated with values of ρ (visit duration to transition times) as follows: $\rho \in \{0.25, 0.5, 1.0, 2.0, 2.25, 3.0\}$. All visit durations were set at 20 minutes, consequently problems have a vehicle speed ranging from 12km/h to 72km/h. Fig. 7 shows scatters of λ computed for the first (λ_{first}) and for the best solutions (λ_{best}) for each particular instance in an ensemble.

In (Beck, Prosser, & Selensky 2002) it was argued that higher ρ's will tend to favour scheduling techniques because

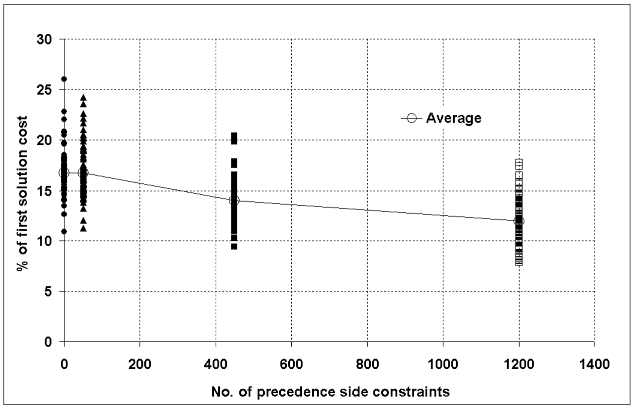

Figure 6: Improvement of the first solution by the routing technique against the number of precedence constraints between activities, $\rho = 1.0$, $\sigma = 0.9$, $p = 1.0$. First solutions were obtained by the scheduling technique.

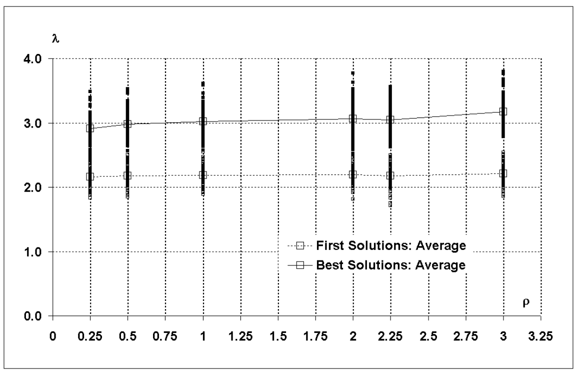

Figure 7: λ_{best} and λ_{first} against ρ, visit duration to transition time. All other parameters take default values: $p = 1.0$, $\sigma = 0.9$.

large values of ρ correspond to small transition times compared to operation durations, typical of scheduling problems. Our results demonstrate, however, that in isolation this parameter does not bring about such a performance difference and routing technology continues to outperform the scheduling technique. In fact, we see that even from a first solution the routing technology is typically twice as good as the scheduling technology, and subsequent search allows the routing technology to dominate scheduling by a factor of three. This is not what we expected; we expected large ρ to correspond to short setup costs, typical of scheduling, and thus more suitable to scheduling technology.

Optimization Criterion (c)

We now solve 100 pure VRPs using two different optimization criteria: minimize total travel (i.e. from a scheduling perspective, minimize the sum of setups), and minimize makespan (in VRP parlance, complete all the visits as early as possible). In Figure 8 we have a scatter plot, on the left λ_{best} and λ_{first} using the minimization of travel, and on the right λ_{best} and λ_{first} where the criterion is minimizing makespan. We see a large difference in behaviour. When minimizing travel (scatter on the left) the routing technology is consistently 2 to 3 times better than the scheduling technology. When we switch to minimizing makespan (scatter on the right) the technologies compete, i.e. λ_{best} and λ_{first} are always about 1. These results are particularly dramatic given that all the other parameters are that of a standard VRP: the only difference is the optimization criterion.

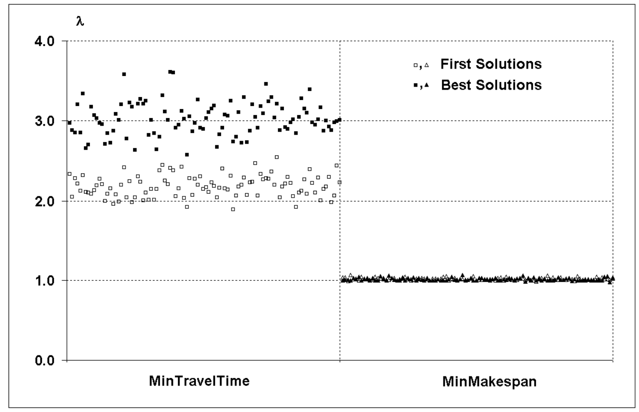

Figure 8: Scatter of λ_{best} and λ_{first} for 100 pure VRPs. On the left, minimize travel time, and on the right minimize makespan.

Why is there such a difference when we change optimization criterion? One answer might be the underlying techniques used in the scheduling technology. When minimizing a maximum function (such as makespan), the upper bound can be directly propagated on the completion time of each operation/visit, and the domains of possible start times can be effectively tightened. However, when a sum function (e.g., sum of transition times) is being optimized, the respective constraint has to be considered at each stage of search simultaneously with resource constraints (Baptiste, Le Pape, & Nuijten 2001) thus making it harder to achieve large domain reductions. The scheduling technology we used is built to exploit the powerful constraint propagation that is possible using such criterion as makespan, whereas the routing technology is not. We might therefore expect when we have a VRP, and the goal is to complete all visits as soon as possible, we might as well solve the VRP as a scheduling problem.

Slack (σ)

To experiment with slack, we generate VRPs with normalized slack $\sigma \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. All other parameters take default values. As σ increase the amount of slack increases, i.e. time windows get larger with respect to visit durations.

Fig. 9 shows λ_{best} and λ_{first} against increasing normalized slack σ. As slack increases we see that routing technology improves relative to scheduling technology. Conversely,

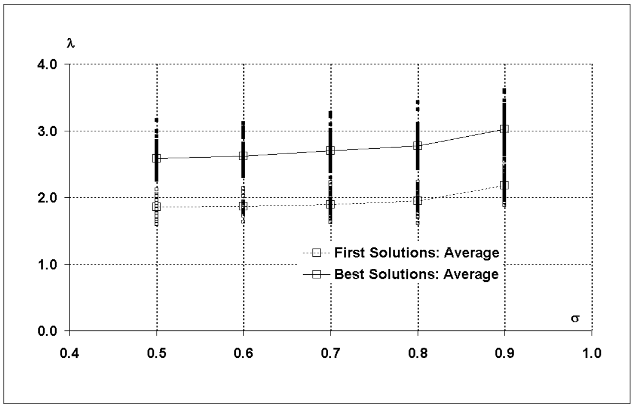

Figure 9: $\lambda = \lambda(\sigma)$, $\rho = 1.0$, $p = 1.0$. First solutions were obtained by the scheduling and routing technique independently.

when slack decreases it appears that scheduling technology is improving, yet continues to be dominated by the routing technology. We might conclude from this that as time slack decreases, and problems become more temporally constrained, we might incorporate scheduling technology into the routing technology.

Summary of Results

We have varied five different parameters, each in isolation. When we increased the specialization of the fleet, we discovered that the routing technology failed to produce a solution. However, the scheduling technology did find a solution, and this solution could then be improved by the routing technology. Again, routing technology failed when presented with problems with only a modest number of precedence constraints. Scheduling technology came to the rescue, producing an initial solution that could again be improved by routing technology. In both of these scenarios, we might think of the scheduling technology as giving us a "smart start". However, as we increased the number of precedence constraints the scope for improvement diminished.

When we increase the speed of vehicles, essentially compressing the routing problem into a smaller space, we expected the VRP to become more like a scheduling problem with short transition times. This didn't happen. The VRP appeared to continue to behave as a VRP. Therefore short travel distances, do not appear to detract from VRP's essential features. We might take this as good news, and expect that the technology will perform well in urban as well as suburban routing problems.

The optimization criterion had a profound effect. When we want to minimize travel, VRP technology was our choice, but when we want to do all our visits as soon as possible (i.e. minimize makespan) the scheduling technology was a clear winner. One explanation for this is how the specialized propagation within scheduling technology exploits the optimization criterion.

Slack is a property of both VRPs and scheduling problems alike. Nevertheless, as we reduced slack scheduling technology appeared to improve relative to routing technology, but was still dominated by the routing technology.

We can also view these results from the perspective of the solving technology. We observe that the savings heuristic used to find a first solution for VRP solving seems particularly sensitive to impurities in a VRP. Reducing the number of resource alternatives or adding precedence constraints resulted in many problems for which the savings heuristic could not generate a feasible solution. This is consistent with previous findings in adding side constraints to VRPs (Kilby, Prosser, & Shaw 2000). In contrast, GLS is able to improve solutions even under parameter settings where the savings heuristic failed. None of our problem sets represented pure scheduling problems and, therefore, we cannot make strong claims about the robustness of the scheduling technology used. This remains for future work.

Conclusions and Future Work

In the paper, we considered the vehicle routing and shop scheduling problems and carried out an empirical study to develop an understanding of the problem characteristics that contribute to the performance of standard solving technologies. We identified five such parameters: vehicle specialization, operation duration to travel time ratio, the presence or absence of complex temporal relationships between operations, the optimization criterion, and slack.[3]

Our experiments show that the most important characteristics that make the routing technology superior to the scheduling technology when applied to the vehicle routing problem are the optimization criterion (minimize total travel time), the openness of the problem (i.e., the precedence constraints among operations) and low resource specialization. Based on this analysis, we might expect routing technology to perform well on open shop scheduling problems. Conversely, as our routing problems become richer, by adding precedence constraints, reducing slack, and specializing the fleet, scheduling technology becomes more useful.

In our study we use existing VRP and scheduling technology found in commercially available tools. This may appear naive. However, we believe that this is a critical feature of our study. The current research literature boasts a number of powerful approaches for solving combinatorial optimization problems (e.g., constraint programming, mixed integer programming, SAT solvers, etc.). These approaches, however, require significant skill and experience to apply to their full potential. We believe that this skill and experience is a major barrier to the more widespread use of this technology and therefore are researching ways to reduce it. As argued in the introduction, one such approach is to attempt to reuse available solution technology "out of the box". In this study, we are not interested in the algorithmic customizations we could make to scheduling or VRP technology to allow it to solve "impure" problems better. Rather we are interested in the problem characteristics and their on impact existing

[3]We do not claim that these are the only parameters. For example, one of our reviewers suggested that resource capacity might also be considered as parameter.

solution techniques. One of our long term goals is to automate the process of problem analysis to be able to suggest appropriate solution technology. The identification of the relationship between problem characteristics and search performance is an important step in this direction.

To this point, the study has been one-sided, asymmetrical. We have started with VRPs and mutated these, modeled them as VRPs and as scheduling problems, and then solved these problems with the technology appropriate for that representation. There should be another side to this study. We should generate scheduling problems, most probably job-shop scheduling problems (JSPs). These could then be represented and solved as JSPs, and represented and solved as VRPs. We might take JSPs and increase the number of resources available to each operation, vary transition times, relax precedence constraints, vary slack, and again change the optimization criterion. Such a study will naturally complement the work reported here.

References

Baptiste, P.; Le Pape, C.; and Nuijten, W. 2001. *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*. Kluwer Academic Publishers.

Beck, J. C., and Fox, M. S. 2000. Dynamic problem structure analysis as a basis for constraint-directed scheduling heuristics. *Artificial Intelligence* 117(1):31–81.

Beck, J. C., and Refalo, P. 2002. A hybrid approach to scheduling with earliness and tardiness costs. *Annals of Operations Research*. in press.

Beck, J.; Davenport, A.; and Fox, M. 1997. Five pitfalls of empirical scheduling research. In *3rd Int. Conference on Principles and Practice of Constraint Programming (CP'97)*.

Beck, J.; Prosser, P.; and Selensky, E. 2002. On the reformulation of vehicle routing probelms and scheduling problems. In *LNAI 2371, Proceedings of the Symposium on Abstraction, Reformulation and Approximation (SARA 2002)*, 282–289.

Brind, C.; Muller, C.; and Prosser, P. 1995. Stochastic techniques for resource management. *BT Technology Journal* 13(1):55–65.

Cesta, A.; Oddi, A.; and Smith, S. 2000. A constraint-based method for project scheduling with time windows. *Journal of Heuristics* 8(1):109–136.

Clarke, G., and Wright, J. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research* 46:93–100.

Davenport, A., and Beck, J. 1999. An investigation into two approaches for constraint directed resource allocation and scheduling. In *INFORMS*.

DeBacker, B.; Furnon, V.; Shaw, P.; Kilby, P.; and Prosser, P. 2000. Solving vehicle routing problems using constraint programming and metaheuritics. *Journal of Heuristics* 6:5001–523.

Fink, E. 1995. Systematic approach to the design of representation-changing algorithms. In *Symposium on Abstraction, Reformulation and Approximation (SARA)*.

Fink, E. 1998. How to solve it automatically: Selection among problem-solving methods. In *4th Int. Conference on Artificial Intelligence Planning Systems*.

Focacci, F.; Laborie, P.; and Nuijten, W. 2000. Solving scheduling problems with setup times and alternative resources. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*.

Fox, M. 1983. *Constraint-Directed Search: A Case Study of Job Shop Scheduling*. Ph.D. Dissertation, Carnegy Mellon University, Intelligent Systems Laboratory, The Robotics Institute, Pittsburgh, PA.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.

Kilby, P.; Prosser, P.; and Shaw, P. 2000. A comparison of traditional and constraint-based heuristics methods for vehicle routing problems with side constraints. *Constraints* 5(4):389–414.

Laborie, P. 2001. Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results. In *Proceedings of the 6th European Conference on Planning (ECP01)*.

Le Pape, C. 1994. Implementation of Resource Constraints in ILOG SCHEDULE: A Library for the Development of Constraint-Based Scheduling Systems. *Intelligent Systems Engineering* 3(2):55–66.

Nuijten, W. P. M. 1994. *Time and resource constrained scheduling: a constraint satisfaction approach*. Ph.D. Dissertation, Department of Mathematics and Computing Science, Eindhoven University of Technology.

Selensky, E. 2001. On mutual reformulation of shop scheduling and vehicle routing. In *Proceedings of the 20th UK PLANSIG*.

Smirnov, Y., and Veloso, M. 1996. Efficiency competition through representation changes: Pigeonhole principle versus linear programming relaxation. In *5th Int. Conference on Principles of Knowledge Representation and Reasoning (KR'96)*.

Smith, S., and Cheng, C. 1993. Slack based heuristics for constraint satisfaction scheduling. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, 139–144.

Solomon, M. 1987. Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. *Operations Research* 35:254–365.

Toth, P., and Vigo, D. 2002. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications.

Voudouris, C., and Tsang, E. 1998. Guided Local Search. *European Journal of Operational Research* 113(2):80–110.

Walsh, T. 2000. Reformulating propositional satisfiability as constraint satisfaction. In *Symposium on Abstraction, Reformulation and Approximation (SARA)*.