

Separating the Inventory Slack Routing Problem

Jeffrey W. Herrmann

The
Institute for
Systems
Research



A. JAMES CLARK
SCHOOL OF ENGINEERING

ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.

ISR is a permanent institute of the University of Maryland, within the A. James Clark School of Engineering. It is a graduated National Science Foundation Engineering Research Center.

www.isr.umd.edu

Separating the Inventory Slack Routing Problem

Jeffrey W. Herrmann
A. James Clark School of Engineering
2181 Martin Hall
University of Maryland
College Park, MD 20742
301-405-5433
jwh2@umd.edu

Abstract

In practice, when faced with a complex optimization problem, human decision-makers often separate it into subproblems and then solve each subproblem instead of tackling the complete problem. This paper describes a study in which different approaches for separating the Inventory Slack Routing Problem (a complex vehicle routing problem) were simulated. A random search was used to simulate how a bounded rational human decision-maker would solve each subproblem. The results show that the structure of the separation and the objectives used in each subproblem can significantly affect the quality of the solutions that are generated. This suggests that organizations should consider and assess how they separate complex optimization problems so that their limited decision-making resources can be employed efficiently and effectively.

Introduction

In practice, human decision-makers often separate a complex optimization problem into subproblems and then solve each subproblem instead of tackling the complete problem. This approach is a natural strategy given the constraints that human decision-makers have. As part of an ongoing study of the effectiveness of this strategy, this paper considers a specific optimization problem and presents the results of a simulation study that modeled a hypothetical bounded rational decision-maker and evaluated the quality of the solutions that this decision-maker found

using different separations. The study was intended to provide insights into the relationship between the quality of the solutions and the structure of the separations.

In particular, the paper studies the Inventory Slack Routing Problem (ISRP), which considers how to deliver inventory from a central depot to a number of demand sites as early as possible given the availability of material at the depot over time (Montjoy and Herrmann, 2012). A key constraint is that a solution must specify sufficient deliveries to every site so that it receives the total amount required. A “slack” measure denotes how early a delivery reaches its destination before it is needed. Slack is the difference in time between the expected time when all previously delivered material will be exhausted and the time that the delivery occurs. Solving the problem with a standard total travel time or total cost objective function would not adequately address the goal of delivering material as early as possible.

A slack value can be calculated for each delivery. Maximizing the sum of these values could yield an inequitable solution in which some deliveries have a large slack (and the sites have excess inventory) and other deliveries have little (or negative) slack (and the sites have little or no inventory). Thus, the ISRP objective is to maximize the minimum slack, which achieves the primary objective of ensuring that material arrives in a timely matter while also equitably allocating material to the sites.

The ISRP involves assigning sites to each vehicle and sequencing them to create routes, determining how many trips each vehicle will make and when each trip will start, and determining the amount of material to deliver to each site on each trip.

The remainder of the paper proceeds by reviewing the concept of bounded rationality, discussing how we modeled bounded rational decision-makers, and describing the separations studied. The paper then describes the instances used in the computational study, the design of

the computational study, and the results of the study before concluding with some insights gained from this work. Additional details about the ISRP can be found in the report by Montjoy and Herrmann (2012).

Bounded Rationality

It is well-known that real-world decision-makers cannot optimize because of limits on their problem-solving capacity. This concept is known as bounded rationality (Simon, 1997a). Bounded rationality reflects the observation that, in most real-world cases, decision-makers have limited information and limited computational capabilities for finding and evaluating alternatives and choosing among them (Simon, 1997b; Gigerenzer et al., 1999; March and Simon, 1993). A decision-maker cannot perfectly evaluate the consequences of the available choices. This prevents complete and perfect optimization. The study of procedures besides optimization to choose between alternatives has been defended by appealing to bounded rationality, as in the case of pairwise comparisons of a finite set of alternatives (Dym et al., 2002).

Satisficing and fast and frugal heuristics are two models of bounded rationality (Gigerenzer et al., 1999). Bounded rational decision-makers may search until they find something that meets their requirements (satisficing), or they may use fast and frugal heuristics that search a limited set of objects and information and make choices using rules that are easy to compute (and therefore quick). This study considers only this second type of bounded rationality.

Gurnani and Lewis (2008) studied collaborative, decentralized design processes in which the models of the individual decision-makers (the designers) were based on the ideas of bounded rationality. In their model, the value chosen by each designer was determined by randomly sampling from a distribution around the (locally) “optimal” solution. This model was meant to

represent the mistakes that designers make due to bounded rationality. Their results showed that incorporating bounded rationality led to more desirable solutions in a collaborative, decentralized design process in which the designers had different objectives and no way to coordinate their activities.

This study took a different approach to modeling bounded rationality. To motivate this approach, this section will first consider models of organizational decision-making and problem solving and then consider cognitive models of human problem solving.

When tackling a complex optimization problem, the decision-maker needs to determine values for various aspects of the solution. That is, he must make a decision. In practice, he does not have complete information about the available alternatives and their impact on overall performance. Thus, he must employ some process for generating and evaluating alternatives.

March and Simon (1993) describe the general characteristics of human problem-solving in organizational decision-making. The first characteristic is that making a complex decision involves making a large number of small decisions. The second characteristic is that problem-solving has a hierarchical structure in which solving any problem goes through phases that, in turn, require solving more detailed subproblems. The general concept of separation is related to these two characteristics. The third characteristic is that problem-solving consists of searching for possible solutions (cf. Nutt, 2005). The fourth characteristic is that problem-solving includes screening processes that evaluate the solutions that are found. The fifth characteristic is that problem-solving has not only random components (such as finding and evaluating solutions) but also a procedural structure that allows it to yield good solutions. The proposed model of a bounded rational designer is motivated by these last three characteristics.

Wang and Chiew (2008) described human problem solving as a higher-layer cognitive process that can be considered as a search process, though it requires other cognitive processes such as abstraction, analysis, synthesis, and decision-making. Their model of a generic problem solving process is a search that iteratively generates and evaluates potential solutions.

Thus, we conclude that search is a valid model of bounded rational decision-making.

Herrmann (2010) took a similar approach but considered separations of an engineering design optimization problem as models of progressive design processes. His results indicated that well-designed progressive design processes are the best way to generate profitable product designs.

This paper considers a very different problem but addresses the same issue: how does the separation affect the quality of the solution?

Modeling Bounded Rationality

An important aspect of bounded rationality is that the resources and time available for problem-solving are limited. Consequently, the proposed model of a bounded rational designer incorporates limits that will constrain the amount of time available for the search and the quality of a solution.

Therefore, the separation assessment method used a class of search algorithms that identify and evaluate solutions to model the choices of a bounded rational designer. Each search includes random components (randomly selecting a solution), and the search is limited to a fixed number of solutions. The procedural structure of the model attempts to compensate for the randomness and resource constraints by keeping track of the best solution found so far.

This study used a random sampling search to solve each subproblem. The search is characterized by N , the search effort. Let X be the vector of variables that need to be determined

in that subproblem. In addition, the solutions may need to satisfy one or more constraints C . Let $f(X)$ be the objective function being used to evaluate solutions for that subproblem.

The random sampling search operated as follows:

Do the following step N times: Randomly select X such that X is feasible with respect to C . If $f(X)$ is the best function evaluation found so far, keep X as the best solution found so far. The search returns the best solution found so far.

It is important to note that this search was meant to represent a bounded rational decision-maker. The search should not be compared to state-of-the-art techniques for solving optimization problems. As Gurnani and Lewis (2008) and Herrmann (2010) did, the assessment method modeled the decision-maker's bounded rational choices as a random process. In the study by Gurnani and Lewis (2008), the values chosen by the decision-maker were determined by a random error term. In the approach used here, on the other hand, the values were chosen by a random search process. To model a bounded rational decision-maker who is using a type of fast and frugal heuristic, this search used simple rules to stop the search (when the number of solutions evaluated equals N) and to choose a solution (whether it is better than the best found so far).

ISRP Separations

A separation is a process that solves a sequence of subproblems.

The concept of separation is similar (but not identical) to the idea of decomposition. Both replace a large design optimization problem with a set of subproblems. In a typical decomposition approach, a second-level problem must be solved to coordinate the subproblem solutions in an iterative manner.

Separation, on the other hand, does not require subsequent coordination. It is a decentralized and sequential approach in which a large problem is divided into subproblems. The solution to one subproblem will provide the inputs to one or more subsequent subproblems. However, there is no higher-level problem to coordinate the solution. Note that the separation does not have to be a simple sequence of subproblems; it may have subproblems that are solved in parallel at places. A given separation specifies a partial order in which the subproblems are solved. A different order of subproblems would be a different separation and would lead to a different solution.

The subproblems' objective functions are surrogates for the original problem's objective function. These surrogates come from substituting simpler performance measures that are correlated with the original one, eliminating components that are not relevant to that subproblem, or from removing variables that will be determined in another subproblem.

Like dynamic programming, separation may solve a set of subproblems and use the solution of one problem to solve another. Typically dynamic programming recursively solves a set of subproblems (corresponding to a set of possible states) starting with a trivial subproblem. By contrast, separation does not contain this special recursive structure; therefore, solving a subproblem considers only the decisions that have been made.

Also, despite the similar name, separation is not the same as separable programming, a branch of mathematical programming that concerns nonlinear optimization problems in which the objective function and the constraints are sums of single-variable functions (cf. Stefanov, 2001). Separable programming approaches use a linear program to approximate the original problem and employ a type of simplex algorithm to find a solution. By contrast, separation replaces the original problem with a set of subproblems.

This study considered six separations of the ISRP. Four of the separations had two versions (corresponding to two different objective functions for the routing subproblem). Figures A1 and A2 in the Appendix include visual representations of the separations. In this study, each subproblem is solved by a random search that simulates the decisions of a bounded rational decision-maker.

Separation 1 considers the routing, scheduling, and allocation variables (denoted by σ , t , and q in Figure A1) simultaneously. In each iteration of the search, routes are constructed by randomly dividing a random permutation of the demand sites. From these routes and a completely proportional allocation of the quantities, the deliver-when-possible algorithm (Montjoy and Herrmann, 2012) is used to determine the schedules (when the vehicles start their trips). For each vehicle, the material available in each delivery is randomly allocated to the demand sites served by that vehicle. This forms a complete solution, and the minimum slack of the solution can be found. When the search is complete, the separation ends with the solution with the greatest minimum slack.

A completely proportional allocation of the material to the demand sites usually yielded a good solution that could be improved by adjusting the delivery quantities. In this study, the search for delivery quantities exploited the quality of the proportional allocations by combining the proportional allocation with randomly determined delivery quantities in two ways. In the first scheme, a predetermined amount of material (either 50% or 5%) was set aside and allocated proportionally (each site received an amount proportional to its total requirements). The remainder was allocated randomly. Each delivery combined the proportionally allocated quantity and the randomly determined quantity. The second scheme first randomly selected a fraction uniformly distributed between 1% and 99%, set aside this fraction of the material for

proportional allocation, and then randomly allocated the remainder. These schemes were used in all of the separations except Separation 2.

Separation 2 also considers the routing, scheduling, and allocation simultaneously, but the allocation is not randomly determined. In each iteration of the search, routes are constructed by randomly dividing a random permutation of the demand sites. From these routes and a completely proportional allocation of the quantities, the deliver-when-possible algorithm is used to determine the schedules. This forms a complete solution, and the minimum slack of the solution can be found. After the search completes, the separation ends with the solution with the greatest minimum slack.

Separation 3 first determines the routes, then creates a schedule, and then allocates material. There are two versions of Separation 3 (3A and 3B); they use different objective functions for evaluating the routes. First, routes are constructed by randomly dividing a random permutation of the demand sites. In Separation 3A, the routes are evaluated by calculating the round-trip time required for each route; the quality of a set of routes is the greatest trip time (denoted by y in Figure A1). The search finishes with the routes with the smallest greatest trip time. In Separation 3B, the routes are evaluated by calculating the time that every delivery is completed; the quality of the route is the greatest delivery time (denoted by w in Figure A1). The search ends with the routes with the smallest greatest delivery time.

After selecting a route, and with a completely proportional allocation of the quantities, the separation uses the deliver-when-possible algorithm to determine the schedules. Then, the separation solves the allocation problem for each and every vehicle (one vehicle at a time). Each iteration of this search randomly allocates the material available in each delivery to the demand sites served by that vehicle. The minimum slack of this partial solution can be found. The

search ends with the quantities that yield the greatest minimum slack for this vehicle. After considering every vehicle, the separation combines these quantities to form a complete solution to the problem.

Separation 4 determines the routes the same way that Separation 3 does. There are two versions of Separation 4 (4A and 4B); for evaluating the routes, they use the objective functions used in Separation 3.

After finding the routes (but before scheduling), the separation solves the allocation problem for all of the vehicles simultaneously. Each iteration of the search randomly allocates the material available in each wave to the demand sites so that each and every demand site is allocated a total that is equal to the total required at that site. Then, with the given routes and allocations, a version of the deliver-when-possible algorithm is used to create the vehicle schedules, and the minimum slack can be found. The search ends with the quantities and schedule that yield the greatest minimum slack.

Separations 5 and 6 first cluster the demand sites and then form a route for each cluster (the number of clusters equals the number of vehicles). (The clusters are denoted by C in Figure A2.) Each separation starts by randomly clustering the demand sites; the clusters form a partition of the demand sites. The maximum travel time between any two demand sites within the same cluster is the objective function; the replication completes this subproblem with the clusters that yield the smallest maximum travel time.

The separation then considers the subproblem of generating routes for the clusters. In each iteration of the search, for each cluster, a random permutation of the sites in the cluster is used to form a vehicle route (in the complete route, the vehicle starts at the depot, visits the sites in order, and then returns to the depot).

There are two versions of Separation 5 (5A and 5B); they use different objective functions for evaluating the routes. In Separation 5A, the routes are evaluated by calculating the round-trip time required for each route; the quality of the route is the greatest trip time. The search ends with the routes with the smallest greatest trip time. In Separation 5B, the routes are evaluated by calculating the time that every delivery is completed; the quality of the route is the greatest delivery time. The search ends with the routes with the smallest greatest delivery time.

Both Separations 5A and 5B, after selecting a set of routes, use the deliver-when-possible algorithm to determine the schedules (with a completely proportional allocation of the material). Then, as Separations 3A and 3B do, each separation solves the allocation problem for each and every vehicle (one vehicle at a time). After considering every vehicle, the separation combines these quantities to form a complete solution to the problem.

There are two versions of Separation 6 (6A and 6B), which differ in the same way that Separations 5A and 5B differ. In Separation 6A, the routes are evaluated by calculating the round-trip time required for each route; the quality of the route is the greatest trip time. In Separation 6B, the routes are evaluated by calculating the time that every delivery is completed; the quality of the route is the greatest delivery time.

Both Separations 6A and 6B, after finding a route (but before scheduling), solve the allocation problem for all of the vehicles simultaneously (as Separations 4A and 4B do). The search ends with the quantities and schedule that yield the greatest minimum slack.

ISRP Instances

Four instances used in the testing described by Montjoy and Herrmann (2012) were used in the assessment of separations. The size of each instance is listed in Table 1.

Table 1. Size of the four instances used in the assessment of the separations. n is the number of demand sites, and V is the number of vehicles.

Instance Name	n	V
Rockville 3	10	5
California 2	20	10
50 PODs	50	25
All MD	189	71

Computational Experiments

The purpose of the computational experiments was to compare the performance of the separations. All of the algorithms were implemented in Matlab and executed using Matlab R2006b on a Dell Optiplex GX745 with Intel Core2Duo CPU 6600 @ 2.40 GHz and 2.00 GB RAM running Microsoft Windows XP Professional Version 2002 Service Pack 3. The computational effort of the simulations increased as the instances grew, but that is not an important performance metric in this work.

The number of iterations in each search was set to two values: $N = 300$ and $N = 1000$. Each separation was run for 25 replications on each of the four ISRP instances. Each replication yielded a complete solution, and we recorded the minimum slack of these solutions. For each separation and each instance, we used this sample to calculate the average minimum slack. For comparing two separations on the same instance, we found the difference in minimum slack for each replication and the 90% confidence interval on the expected difference in minimum slack.

When implementing the separations, it was more efficient to avoid running the same search multiple times. This also reduced some of the variation between the searches. The solutions to the first subproblem for Separation 3A were used in Separation 4A. The solutions to the first subproblem for Separation 3B were used in Separation 4B. The solutions to the first subproblem for Separation 5A were used in Separations 5B, 6A, and 6B. The solutions to the

second subproblem in Separation 5A were used in Separation 6A. The solutions to the second subproblem in Separation 5B were used in Separation 6B.

Results

The performance of each separations is assessed by the quality of the solutions that it generates. In general, the solution quality was not significantly affected by the rule used to set aside some quantity for proportional allocation (using 50%, using 5%, or using a random amount). For example, as shown in Table 2, the only significant difference for the 50 PODs instance was between Separations 1, 5A, and 5B. For Separations 3A, 3B, 4A, 4B, 6A, and 6B, the proportional allocation rule made no significant difference.

Table 2. For the 50 PODs instance, 90% confidence intervals on the mean difference between the min slack of solutions generated by setting aside a random amount and the min slack of solutions generated by setting aside a fixed 50% or 5% of the material for proportional allocation. Note that Separation 2 is not included because it always proportionally allocated all of the material. (Positive values correspond to larger min slack values for the solutions generated by setting aside a random amount.) An asterisk (*) denotes a significant difference.

Separation	50% $N = 300$	50% $N = 1000$	5% $N = 300$	5% $N = 1000$
1	(-2.1 ± 9.0)	$(16.4 \pm 7.8)^*$	(9.1 ± 9.2)	$(27.6 \pm 8.9)^*$
3A	(1.0 ± 12.4)	(-3.5 ± 8.6)	(-1.6 ± 12.6)	(-9.6 ± 10.3)
3B	(9.2 ± 10.3)	(6.4 ± 10.6)	(2.0 ± 9.7)	(2.3 ± 10.7)
4A	(1.7 ± 10.0)	(-3.8 ± 6.6)	(3.8 ± 11.2)	(-1.5 ± 8.4)
4B	(4.4 ± 11.1)	(-1.5 ± 9.1)	(9.4 ± 9.7)	(2.7 ± 8.3)
5A	$(3.2 \pm 1.5)^*$	$(4.5 \pm 2.4)^*$	$(-5.0 \pm 2.4)^*$	$(-3.9 \pm 1.7)^*$
5B	(1.0 ± 2.9)	$(6.0 \pm 1.8)^*$	$(-6.9 \pm 2.8)^*$	$(-2.7 \pm 1.8)^*$
6A	(-12.1 ± 31.7)	(2.7 ± 29.5)	(-13.8 ± 29.9)	(-0.6 ± 27.7)
6B	(-9.1 ± 32.4)	(-0.2 ± 31.8)	(-8.0 ± 30.2)	(-2.2 ± 29.6)

Table 3. The 90% confidence intervals on the mean difference between the min slack of solutions generated by Separation 2 and the min slack of solutions generated by Separation 1. (Positive values correspond to larger min slack values for the solutions generated by Separation 2.) An asterisk (*) denotes a significant difference.

Instance	Varying $N = 1000$	50% $N = 1000$	5% $N = 1000$
Rockville 3	$(-2.1 \pm 2.0)^*$	(2.2 ± 2.5)	$(5.1 \pm 2.2)^*$
California 2	(-3.3 ± 3.3)	(0.2 ± 3.3)	(0.2 ± 3.9)
50 PODs	(-8.9 ± 9.5)	(7.5 ± 11.2)	$(18.7 \pm 11.3)^*$
All MD	$(-18.6 \pm 9.6)^*$	(8.4 ± 15.7)	$(80.6 \pm 17.0)^*$

Separation 2 always proportionally allocated all of the material but is otherwise similar to Separation 1. When we compare the performance of these separations, for $N = 1000$, it appears that Separation 1 generates solutions that have slightly better min slack than those generated by Separation 2 when the fraction allocated proportionally is randomly chosen. When the fraction allocated proportionally is fixed at 50%, the solutions generated by Separation 1 are not significantly better or worse than those generated by Separation 2. When the fraction allocated proportionally is fixed at 5%, the solutions generated by Separation 1 have significantly less min slack than those generated by Separation 2.

When the fraction allocated proportionally is randomly chosen, the fractions in the best solutions found by Separation 1 tended to be greater than 50%. For the Rockville 3 instance, 20 of the 25 best solutions used fractions greater than 50%; for the California 2 instance, 13; for the 50 PODs instance, 15; and, for the All MD instance, 21. Thus, it appears that, for Separation 1, larger fractions led to higher quality solutions.

To compare the separations we will consider the solutions in which the fraction set aside for proportional allocation was 5% and $N = 1000$ iterations for each subproblem.

For the Rockville 3 instance, the solutions generated by Separation 3B had the largest average slack. They were significantly better than those generated by any other separation.

For the California 2 instance, because the minimum slack occurred in the first wave of deliveries, which the delivery quantities could not change, Separations 3A and 4A and Separations 3B and 4B performed identically. Separations 5A and 6A and Separations 5B and 6B should have performed identically, but they were not run at the same time. The solutions generated by Separations 3B and 4B were significantly better than those generated by any other separation.

For the 50 PODs instance, the solutions generated by Separations 3A, 3B, 4A, and 4B were not significantly different. The solutions generated by Separation 3B were significantly better than those generated by Separations 1, 2, 5A, 5B, 6A, and 6B.

For the All MD instance, the solutions generated by Separations 3A and 3B were not significantly different. The solutions generated by Separation 3B were significantly better than those generated by Separations 1, 2, 4A, 4B, 5A, 5B, 6A, and 6B.

Except for California 2, in which the minimum slack is determined by only the routes, the quality of the solutions generated by Separation 1 (relative to those generated by Separation 3B) degrades as the instances get larger. For the Rockville 3 instance, the 90% confidence interval on the expected difference in min slack was (17.26, 20.68). For the 50 PODs instance, the 90% confidence interval on the expected difference in min slack was (49.63, 70.56). For the All MD instance, the 90% confidence interval on the expected difference in min slack was (96.46, 122.04).

In general, when generating the routes, minimizing the maximum delivery time was a better strategy than minimizing the maximum route time. That is, Separation 3B was better than Separation 3A, Separation 4B was better than Separation 4A, Separation 5B was better than Separation 5A, and Separation 6B was better than Separation 6A. The difference was significant

for the instances Rockville3 and California2 but not for the instances 50 PODs and All MD.

Table 4 shows the differences for the Rockville3 instance.

Table 4. For the Rockville 3 instance, 90% confidence intervals on the mean difference between the min slack of solutions generated by separations that minimized the maximum delivery time and the min slack of solutions generated by separations that minimized the maximum route time. (Positive values correspond to larger min slack values for the solutions generated by Separations 3B, 4B, 5B, and 6B.) All of the differences were significant.

Separations	Varying $N = 1000$	50% $N = 1000$	5% $N = 1000$
3B - 3A	(3.0 ± 2.0)	(4.3 ± 2.1)	(4.5 ± 1.9)
4B - 4A	(7.0 ± 1.9)	(6.1 ± 1.4)	(4.5 ± 2.1)
5B - 5A	(2.3 ± 0.7)	(2.5 ± 0.5)	(2.3 ± 0.5)
6B - 6A	(1.5 ± 1.3)	(3.3 ± 1.7)	(3.1 ± 2.2)

The variation in the quality of the solutions generated by Separations 5A, 5B, 6A, and 6B was much larger than the variation in the quality of the solutions generated by the other separations. This results in part from the variation in the quality of the routes found by these cluster-first separations. As shown in Table 5, the average max delivery time in the routes generated by Separations 5A, 5B, 6A, and 6B was larger than the average max delivery time of the solutions generated by the Separations 3A, 3B, 4A, and 4B; moreover, the variation (measured by the coefficient of variation) was larger as well. Separations 5A, 5B, 6A, and 6B only sometimes generated routes nearly as good as those generated by the Separations 3A, 3B, 4A, and 4B; at other times, the max delivery times were much larger, which naturally reduced slack.

Table 5. The average (and coefficient of variation) of the max delivery time of the routes generated by separations 3A, 3B, 4A, 4B, 5A, 5B, 6A, and 6B. The coefficient of variation is the standard deviation divided by the average value. Note that pairs of separations generated the same routes and are therefore grouped together in this table.

Separations	Rockville 3	California 2	50 PODs	All MD
3A, 4A	95 (0.101)	169 (0.065)	141 (0.053)	305 (0.059)
3B, 4B	81 (0.018)	151 (0.029)	138 (0.045)	296 (0.031)
5A, 6A	97 (0.154)	177 (0.110)	163 (0.155)	353 (0.130)
5B, 6B	94 (0.149)	164 (0.123)	158 (0.173)	337 (0.136)

Summary and Conclusions

This paper presented the results of simulating various separations of a complex vehicle routing problem that includes inventory management as well.

Note that these results are not meant to imply that decision-makers should use any of these algorithms explicitly to solve complex optimization problems. They are models of how a bounded rational decision-maker makes choices, and their purpose is to help assess the quality of different separations.

First, the structure of a separation and the objectives of its subproblems affect the quality of the solutions generated. In the case of the ISRP, using the maximum delivery time (instead of the total route time) identifies routes that yield better solutions for the ISRP. Also, searching for a complete solution (as Separation 1 does) tends to generate solutions that are not as good as those generated by finding a good route first; likewise, clustering first also tends to generate inferior solutions. Thus, organizations should assess solution quality to guide the design of their decision-making processes.

Second, although there is no automated way to separate complex optimization problems into subproblems, one strategy is to examine the coupling between the decision variables. In many cases, the variables are loosely coupled (Simon, 1981). There will be tightly-coupled groups of variables that should be set at the same time, but the different groups can be separated into different decisions. For instance, in the ISRP, the route variables are tightly coupled to each other but loosely coupled with the delivery quantities. Thus, it would be difficult to design a separation that sets only some of the routes but not the others.

Finally, because human decision-makers have limitations and cannot optimize, well-designed separations are the best way to find quality solutions and make decisions. Trying to set

everything all-at-once will lead to problems that are too large to solve well because the large solution space makes it difficult for the bounded rational decision-maker to approach an optimal solution.

Acknowledgements

Adam Montjoy wrote some of the computer code and constructed the instances used in this work.

References

- Dym, Clive L., William H. Wood and Michael J. Scott, 2002, "Rank Ordering Engineering Designs: Pairwise Comparison Charts and Borda Counts," *Research in Engineering Design*, 13(4), pp. 236-242.
- Gigerenzer, Gerd, Peter M. Todd, and the ABC Research Group, 1999, *Simple Heuristics That Make Us Smart*, Oxford University Press, New York.
- Gurnani, Ashwin, and Kemper Lewis, 2008, "Collaborative, Decentralized Engineering Design at the Edge of Rationality," *Journal of Mechanical Design*, 130(12), 121101.
- Herrmann, Jeffrey W., "Progressive Design Processes and Bounded Rational Designers," *Journal of Mechanical Design*, August 2010, Volume 132, Issue 8, 081005 (8 pages).
doi:10.1115/1.4001902
- March, James, and Herbert Simon, 1993, *Organizations*, second edition, Blackwell, Cambridge, Massachusetts.
- Montjoy, Adam, and Jeffrey W. Herrmann, "Optimizing Urgent Material Delivery by Maximizing Inventory Slack," Technical Report 2012-6, Institute for Systems Research, University of Maryland, College Park, June 12, 2012. Available online at <http://hdl.handle.net/1903/12499>.

- Nutt, Paul C., 2005, "Search During Decision-Making," *European Journal of Operational Research*, 160, pp. 851-876.
- Simon, Herbert A., 1981, *The Sciences of the Artificial*, second edition, MIT Press, Cambridge, Massachusetts.
- Simon, Herbert A., 1997a, *Models of Bounded Rationality*, Volume 3, The MIT Press, Cambridge, MA.
- Simon, Herbert A., 1997b, *Administrative Behavior*, fourth edition, The Free Press, New York.
- Stefanov, Stefan M., 2001, *Separable Programming: Theory and Methods*, Kluwer Academic Publishers, Dordrecht, Netherlands.
- Wang, Y., and Chiew, V., 2008, "On the Cognitive Process of Human Problem Solving," *Cognitive Systems Research*, doi:10.1016/j.cogsys.2008.08.003.