FACULTEIT ECONOMIE
EN BEDRIJFSKUNDE

HOVENIERSBERG 24
B-9000 GENT
Tel. : 32 - (0)9 – 264.34.61
Fax. : 32 - (0)9 – 264.35.92

WORKING PAPER

A Comparison and Hybridization of Crossover Operators for the Nurse Scheduling Problem

Broos Maenhout [1]
Mario Vanhoucke [2]

February 2006

2006/366

[1] Faculty of Economics and Business Administration, Ghent University, Gent, Belgium
[2] Faculty of Economics and Business Administration, Ghent University, Gent, Belgium
Operations & Technology Management Centre, Vlerick Leuven Gent Management School, Gent, Belgium
broos.maenhout@ugent.be • mario.vanhoucke@ugent.be

D/2006/7012/11

ABSTRACT

In this paper, we present a hybrid genetic algorithm for the well-known nurse scheduling problem (NSP). The NSP involves the construction of roster schedules for nursing staff in order to maximize the quality of the roster schedule and to minimize the violations of the minimal coverage requirements subject to various hard case-specific constraints. In literature, several genetic algorithms have been proposed in literature to solve the NSP under various assumptions. The contribution of this paper is twofold. First, we extensively compare the various crossover operators and test them on a standard dataset in a solitary approach. Second, we propose several options to hybridize the various crossover operators.

Keywords: *meta-heuristics; hybridization; nurse scheduling*

1 Introduction

Many service and industrial organizations are confronted with personnel scheduling problems. For most of these organizations, it is of critical importance to have suitably qualified staff on duty at the right time since this is a large determinant of service organization efficiency and customers' requirements satisfaction (Thompson, 1995; Felici and Gentile, 2004). Hence, a broad research attention has been given in literature to a great variety of personnel rostering applications (see Ernst et al., 2004a; Ernst et al., 2004b). In general, personnel scheduling is the process of constructing duty timetables for staff to meet a time-dependent demand while encountering specific workplace agreements and attempting to satisfy individual work preferences. The particular characteristics of different industries result in quite diverse rostering models which lead to the application of very different solution techniques to solve these models. Typically, personnel scheduling problems are highly constrained and complex optimization problems (Ernst et al., 2004b; Glover and McMillan, 1986).

In this paper, we present a procedure to solve the nurse scheduling problem (NSP) which involves the construction of occupation rosters for nursing staff over a pre-defined period. Problem descriptions and models vary drastically and depend on the characteristics and policies of the particular hospital. Since the nurse scheduling problem has this multitude of formulations in terms of hard and soft constraints and objective function possibilities, many exact and heuristic procedures have been proposed to solve the NSP. Recent literature surveys (Cheang et al., 2003; Burke et al., 2004) give an overview of all these procedures, and mention simulated annealing, tabu search and genetic algorithms as popular meta-heuristics for the NSP.

In constructing a nurse roster over a pre-defined scheduling horizon, nurses need to be assigned to shifts in order to meet the minimal coverage constraints and other case-specific constraints and to maximize the quality of the constructed timetable. In the modern work environment, the quality of a personnel roster is measured in terms of satisfying the individual nurses' preferences (Ernst, 2004b). Hence, quantifying these preferences in the objective function guarantees the quality of the nurse roster over the scheduling horizon and provides fairness between the nurses (Warner, 1976). The coverage constraints express the required nurses per shift and per day, and are inherent to any shift scheduling problem. The coverage constraints are handled as soft constraints that can be violated at a certain penalty cost expressed in the objective function. Case-specific constraints (determined by personal time requirements, specific workplace conditions, national legislation, etc…) are handled as hard constraints, for which no violation is possible whatsoever. The objective is thus to minimize the nurses' preferences, expressed as the aversion to work a particular shift on a particular day, and to obtain a feasible schedule as much as possible subject to different case-specific (hard) constraints. The problem is known to be NP-hard (Osogami and Imai, 2000).

In this paper, we investigate the potential of hybridization of crossover operators in a genetic algorithm to solve the nurse scheduling problem. In section 2, we give an overview of the existing crossover operators in literature. In section 3, we compare these crossover operators on a standard dataset NSPLib and combine them to construct a hybrid genetic algorithm for the nurse scheduling problem. Hansen and Mladenovic (2001) have successfully explored the idea of variable neighbourhood search. This concept consists in systematically changing neighbourhoods during the search and, hence, favours the hybridisation of search mechanisms. In their overview papers, Cheang et al. (2003) and Burke et al. (2004) mention the promising potential with respect to hybridization of meta-heuristic approaches for the nurse scheduling problem. In section 4, conclusions are drawn and directions for future research are given.

2 Our general genetic algorithm for the NSP

2.1 The GA framework for the NSP

Genetic Algorithms (GAs), established by Holland (1975), are adaptive heuristic search algorithms imitating the evolutionary ideas of natural selection and genetic processes. The basic concept of GAs consists in simulating evolutionary processes and the principle of the survival of the fittest. As such, they represent an intelligent exploitation of a random search within a defined search space to solve the problem under study. Hence, GAs have attracted much attention in solving various scheduling problems in many application areas. According to Michalewicz (1995), the application and success of the exploitation of GAs depends on the incorporation of problem specific information. The pseudo-code for the evolutionary framework of our genetic algorithms to solve the NSP is written below.

```
Algorithm GA NSP
    Initialize a Population of Individuals
    Evaluate Fitness of each Individual
    While Stop Criterion not met
        Selection of Individuals to Combine
        Application of Crossover Operator
        Application of Mutation Operator
        Application of Local Search Heuristics
        Evaluation of Fitness of the Newly Created Individuals
        Update Population
    Endwhile
```

The genetic algorithm to solve the NSP starts with the *initialization of a population* of individuals. Solutions are usually generated in a random manner. However, Reeves (1995) and Kapsalis et al. (1993) pointed out that introducing high-quality solutions, obtained from a heuristic technique can help a GA to find good solutions more quickly. Hence, we create the initial solutions by means of an efficient and simple constructive heuristic. The constructive heuristic schedules the nurses in a random sequence taking both preference costs and penalty cost of violating the coverage constraints into account. The scheduling of a particular nurse over the complete planning horizon is conceived as a minimum cost flow problem over a network which embodies all shifts on all days to which a particular nurse can be assigned to. Since not all case-specific constraints (e.g. the maximum number of assignments) can be modelled in the network, it has been implemented by a k-shortest path approach.

In evolutionary heuristics, each generated individual solution is encoded in a data structure, called a chromosome. According to Cheang et al. (2003), the NSP is acquainted with three different representation forms, i.e. the nurse-day view, the nurse-task view and the nurse-pattern view. The nurse-day view and the nurse-task view are close variants of each other and are direct representations of the two-dimensional duty rosters. In both views, the data representation directly reflects respectively the shifts or tasks the nurse is assigned to over the scheduling period. The nurse-pattern view, however, is different from these two other views since the nurse's schedule is indirectly presented in the data representation structure. Aickelin (2004) uses an indirect encoding based on permutations of the nurses and a heuristic decoder that builds schedules from these permutations. The chromosomes serve as input for a *fitness function* which evaluates the quality of the solution encoded in each chromosome. As stated before, the fitness function calculates the total aversion of the nurses with respect to the constructed roster and incorporates a penalty cost each time the coverage constraints are violated. At each evolution cycle, pairs of chromosomes in the current population, called parent chromosomes, are randomly selected which *exchange information* in such a way a new individual is created with attributes of both the parent chromosomes. Depending on the data encoding, different crossover operators have been applied in literature for the NSP. The appropriate crossover operators for the NSP are discussed in section 2.2. Furthermore, each new chromosome can undergo another transformation by assigning random schedules to nurses, called a *mutation*, to prevent convergence during the search process. Analogous to natural selection, the heuristic thrives on the idea that the parent chromosomes will pass their good characteristics on to the newly created solution points. Hence, the algorithm preserves or even improves the good characteristics of the parent chromosomes as the population evolves. The strategy for selecting the parent chromosomes as well as the specific characteristics of the recombination (crossover) and mutation operations all involve randomized choices. The *local search heuristics* explore neighbouring solution points of the newly generated solution point in order to improve both total preference cost and coverage infeasibilities. To that purpose, we implemented the three complementary local search algorithms of Maenhout and Vanhoucke (2005a), each focusing on a different part of the scheduling matrix. The pattern-based local search aims at the optimization of the schedule of a single nurse given the schedules of all other nurses. The day-based local search optimizes a single day of the nurse roster given the assignments of the nurses on all other days. The schedule-based local search focuses on the whole schedule for all nurses by swapping (sub-parts of) schedules between nurses. After this intensification process the fitness function is used to *evaluate the improved new solution point*. The population evolves over time with the entrance of new solutions points and the drop-out of old solutions, improving the quality of the best known solution during the search process. The search heuristic is performed until a maximum number of schedules is evaluated.

2.2 The various crossover operators for the NSP GA

In this section, we give an overview of various crossover operators presented in literature. Since the crossover operators depend on the data representation structure, we distinguish between nurse-task or nurse-day based crossover operators (section 2.2.1) and nurse-pattern view based crossover operators (section 2.2.2). Note that we have adapted and translated each specific crossover operator from literature to the settings and specific assumptions of our NSP GA method under study (see section 1). In doing so, we allow an unambiguous comparison between the different crossover operators, since the nurse scheduling problem is described in literature in various guises with respect to representation form, objective function, hard and soft constraints, etc…. The specific crossover operators which are not suited in our NSP framework are not mentioned and not tested.

2.2.1 Crossover operators for the nurse-task/day data representation

We distinguish between three different types of crossover operators that make use of the nurse-task or nurse-day representation, i.e. nurse-based crossover operators, day-based crossover operators, and specific crossover operators that do not fit completely in the two latter types.

Nurse-based crossover operators

The *nurse-based one-point crossover* (NBOP) operator of Aickelin (2000) randomly selects a crossover point between 1 and the number of nurses, such that the individual nurse schedules before the crossover point are copied from the one parent and the individual nurse schedules after the crossover point are copied from the other parent.
The *nurse-based crossover with tournament selection* (NBTS) is based on Burke et al. (2001) and creates a child schedule that combines the best individual nurse schedules from both parents. The notion of 'best' individual nurse schedule refers to a single nurse's schedule which, in case of removal from the parent solution, would lead to the worst deterioration of its objective function value (preference cost and coverage penalty cost).
The *randomly selected nurse-based crossover* (NBRS) is used by Aickelin (2000), Burke et al. (2001) and Dias et al. (2003) and involves the random construction of a binary vector with a length equal to the number of nurses. The child roster contains a mix of individual nurse schedules copied from one parent (the binary vector displays a 1 on the nurse's position) or the other parent (the binary vector displays a 0 on the nurse's position). Consequently, this process boils down to a complete random selection of the nurses' schedules from both parents.

Day-based crossover operators

The *day-based one-point crossover* (DBOP) of Inoue and Furuhashi (2003) randomly selects a crossover point between 1 and the number of days, such that the sub-schedule before the crossover point is copied from the one parent and the sub-schedule after the crossover point is copied from the other parent.
The *day-based crossover with tournament selection* (DBTS) selects the best days out of the two parents to create a child chromosome. This crossover operator is analogue to the NBTS operator, and has – to the best of our knowledge – not been used in literature.
The *randomly selected day-based crossover* (DBRS) is a multiple point crossover which creates a binary vector with a length equal to the number of days. Consequently, when 1 is indicated in the binary vector, the corresponding day of one of the parents is copied. When 0 is indicated the

corresponding day out of the other parent is copied. This crossover operator is analogue to the NBRS operator, and has – to the best of our knowledge – not been used in literature.

Since this abrupt breaking up of both parent solutions by applying day-based crossover operators can violate the (case-specific) hard constraints, the feasibility of the child chromosome needs to be restored using a repair function. This repair function is based on the k-shortest path approach used to build the initial solution (constructive heuristic) and reinstates the schedule of a single nurse by searching the feasible schedule most resembling to the nurse's schedule constructed by the crossover operator violating hard constraints.

Specific crossover operators

The *single parent specific crossover* (SPSC) of Jan et al. (2000) constructs a child chromosome from one single parent chromosome by selecting the 'worst' nurse schedule and the schedule of another randomly selected nurse. The notion of 'worst' individual nurse schedule refers to a single nurse's schedule which, in case of removal from the parent solution, would lead to the smallest deterioration of its objective function value (preference cost and coverage penalty cost). Two time-related crossover points are selected such that the part of the schedules before the first point and the part after the second point of the selected nurse are swapped to the other nurse while the middle part remains unchanged. Hence, this crossover operator never leads to additional violations of the coverage constraints. The crossover operator of Jan et al. (2000) only allows the construction of feasible individual nurse schedules (with respect to the case-specific constraints), by selecting the time related crossover points carefully. This is always possible since the selection of the first (second) crossover point as the start (end) of the scheduling horizon would lead to a simple copy of the entire individual nurse schedule of the parent solution to the child solution.

The *multi parent specific crossover* (MPSC) of Burke et al. (2001) and Dias et al. (2003) uses time-related crossover points per nurse to create child solutions from two parent solutions. The sub-schedule between the start of the scheduling horizon and the first point and between the second point and the end of the scheduling horizon are copied from the first parent, while the part between the two crossover points is copied from the second parent. Burke et al. (2001) only generates one crossover point per nurse, which is identical to the approach of Dias et al. (2003) with the second crossover point equal to the end of the scheduling horizon. Similar to the SPSC approach, the time-related crossover point are chosen such that only feasible individual nurse schedules can be created.

The *best placed events crossover* (BPEC) of Burke et al. (2001) copies the 'best placed events' from two parents to the child solution for every nurse. The 'best placed events' refer to those assignments that would lead to the worst deterioration in the objective function (in terms of both preference and coverage violation costs) when removed. The child solution needs to be repaired after application of the crossover operator since the abrupt disruption of the parent chromosomes can lead to violations of the hard constraints. Feasibility of the child solution is restored by the same repair function used with the day-based operators.

The various crossover operators are illustrated on an example NSP instance with 5 nurses and a scheduling period of 4 days. We assume that each day consists of three shifts (e.g. early (s_1), day (s_2), night (s_3)) and a free shift (s_4). Since "s_4" is used to refer to a free shift, its daily coverage requirements equal zero. We assume some additional case specific constraints as follows: the number of assignments varies between a minimal value of 3 and a maximal value of 4. The consecutive working shifts vary between a minimal value of 2 and a maximal value of 4. Furthermore, the assignment of nurses to maximal one shift per day and the succession constraints are inherent to continuous personnel scheduling problems. The latter constraint implies forbidden successive assignments between s_3 and s_1, s_3 and s_2 and s_2 and s_1. Two different parent solutions have been displayed in figure 1, which displays the preference matrix (the specific assignments have been encircled) and the minimal coverage requirement (row 'coverage'). The left parent has a total preference cost of 60 and one coverage violation (penalized at a unit penalty cost of 100) as indicated in the 'violation' row. The parent solution right has a total preference cost of 64 and one coverage violation, which leads to a total objective function value cost of 164.

	Parent Solution 1															
	Day 1				Day 2				Day 3				Day 4			
	s_1	s_2	s_3	s_4	s_1	s_2	s_3	s_4	s_1	s_2	s_3	s_4	s_1	s_2	s_3	s_4
Nurse 1	3	2	0	1	0	0	3	6	2	0	4	1	3	5	4	9
Nurse 2	8	0	3	9	2	8	0	7	8	1	4	9	9	6	2	2
Nurse 3	4	5	9	2	9	8	3	4	0	6	1	5	0	0	6	6
Nurse 4	9	4	7	2	2	0	5	6	4	6	1	5	2	2	2	0
Nurse 5	1	9	6	8	7	1	1	2	3	2	8	8	3	0	7	2
Scheduled	3	1	1	0	2	2	1	0	2	2	1	0	1	2	1	1
Coverage	2	1	1	0	1	2	1	0	2	1	1	0	2	1	1	0
Violation	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

	Parent Solution 2															
	Day 1				Day 2				Day 3				Day 4			
	s_1	s_2	s_3	s_4	s_1	s_2	s_3	s_4	s_1	s_2	s_3	s_4	s_1	s_2	s_3	s_4
Nurse 1	3	2	0	1	0	0	3	6	2	0	4	1	3	5	4	9
Nurse 2	8	0	3	9	2	8	0	7	8	1	4	9	9	6	2	2
Nurse 3	4	5	9	2	9	8	3	4	0	6	1	5	0	0	6	6
Nurse 4	9	4	7	2	2	0	5	6	4	6	1	5	2	2	2	0
Nurse 5	1	9	6	8	7	1	1	2	3	2	8	8	3	0	7	2
Scheduled	3	1	1	0	2	2	1	0	2	1	2	0	1	1	2	1
Coverage	2	1	1	0	1	2	1	0	2	1	1	0	2	1	1	0
Violation	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Figure 1. Two parent schedules expressed in the nurse-day representation

Figure 2 illustrates the construction of child solutions from the example parents of figure 1 using our different crossover operators. The middle column displays both parents in a more condensed format (each row displays an individual nurse schedule and each column displays the selected shift for each day) and gives an indication of the assignments subject to the crossover operator. The left column specifies the crossover operator details. The right column shows the newly created child solutions.

Crossover Operator	Parent Solutions		Constructed Child Solutions	
	Parent Solution 1	**Parent Solution 2**	**Child Solution 1**	**Child Solution 2**
NBOP: Two child solutions are constructed based on the indicated assignments in both parent solutions which are chosen based on a randomly chosen crossover point between nurse 2 and nurse 3.	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$
	Parent Solution 1	**Parent Solution 2**	**Child Solution**	
NBTS: A single child solution is constructed based on the indicated assignments in both parent solutions. These assignments are the result of a pairwise comparison between the two parents of the deterioration in the objective function value in case of removal of each nurse's individual schedule.	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_1\ s_1\ s_1$	
	Parent Solution 1	**Parent Solution 2**	**Child Solution 1**	**Child Solution 2**
NBRS: Two child solutions are constructed based on the indicated assignments in both parent solutions which are chosen based on a randomly generated nurse vector 10101.	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_1$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_2$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_1\ s_1\ s_1$
	Parent Solution 1	**Parent Solution 2**	**Child Solution 1**	**Child Solution 2**
DBOP: Two child solutions are constructed based on the indicated assignments in both parent solutions which are chosen based on a randomly chosen crossover point between day 2 and day 3. The child solutions might need a repair function due to case-specific constraints violations (see indicated cells in child solutions).	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_1\ s_1$	$s_3\ s_3\ s_2\ s_3$ $s_1\ s_2\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_2\ s_2\ s_3\ s_4$ $s_1\ s_1\ s_2\ s_2$
	Parent Solution 1	**Parent Solution 2**	**Child Solution**	
DBTS: A single child solution is constructed based on the indicated assignments in both parent solutions which denote the 'best' days in terms of the total objective function value. The child solutions might need a repair function due to case-specific constraints violations (see indicated cells in child solutions).	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_3\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_2\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_2\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_1\ s_2$	
	Parent Solution 1	**Parent Solution 2**	**Child Solution 1**	**Child Solution 2**
DBRS: Two child solutions are constructed based on the indicated assignments in both parent solutions which are chosen based on a randomly generated day vector 1001. The child solutions might need a repair function due to case-specific constraints violations (see indicated cells in child solutions).	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_2\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_2\ s_3\ s_4$ $s_1\ s_1\ s_1\ s_2$	$s_3\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_1$
	Parent Solution 1		**Child Solution**	
SPSC: A single child solutions is constructed based on the indicated assignments of one randomly selected parent solutions. The algorithm selects nurse 2 as the 'worst' nurse while nurse 5 is selected randomly. The two crossover points are randomly selected between day 1 and day 2 (first point) and at the end of the scheduling horizon (second point).	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$		$s_2\ s_2\ s_2\ s_3$ $s_1\ s_2\ s_2\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_1\ s_1\ s_2$	
	Parent Solution 1	**Parent Solution 2**	**Child Solution 1**	**Child Solution 2**
MPSC: Two child solutions are constructed based on the indicated assignments in both parent solutions. These assignments have been chosen based on two crossover points per nurse (nurses 1, 3 and 4 have only one crossover point, which means that the second point equals the start or end of the scheduling horizon).	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_2$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_4$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$
	Parent Solution 1	**Parent Solution 2**	**Child Solution**	
BPEC: A single child solution is constructed based on the indicated assignments in both parent solutions which are the best assignments in terms of the total objective function value. The child solutions might need a repair function due to case-specific constraints violations (see indicated cells in child solutions).	$s_2\ s_2\ s_2\ s_3$ $s_1\ s_1\ s_1\ s_2$ $s_1\ s_1\ s_1\ s_1$ $s_3\ s_3\ s_3\ s_4$ $s_1\ s_2\ s_2\ s_2$	$s_3\ s_3\ s_3\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_3$ $s_1\ s_1\ s_1\ s_1$	$s_3\ s_2\ s_2\ s_3$ $s_1\ s_2\ s_2\ s_4$ $s_1\ s_1\ s_1\ s_2$ $s_2\ s_2\ s_3\ s_4$ $s_1\ s_2\ s_1\ s_1$	

Figure 2. The crossover operators used in a nurse-day/task representation

2.2.2 Crossover operators for the nurse-pattern data representation

Aickelin (2004) uses an indirect genetic algorithm with a nurse-pattern data representation based on a permutation chromosome of nurses and a decoder that builds schedules from these permutations. The best decoder found by Aickelin (2004) schedules one nurse at a time and is designed to consider both the feasibility of the schedule as well as the nurses' preferences. In our paper, we have implemented this approach by scheduling nurses in the order they appear in the chromosome by finding for each nurse the shortest path on a network representing all shifts on all days to which the nurse can be assigned to. The distance value of each arc between two nodes in the network is equal to the sum of the preference cost of assigning the nurse to the day/shift of the terminal node and the negative penalty cost if the coverage constraints are violated for that shift on that day. Since not all case-specific assignments can be modelled in such a network, a k-shortest path approach is used finding the shortest path which is feasible with respect to all case-specific constraints. The indirect genetic algorithm of Aickelin (2004) uses five different crossover operators, as follows:

The *C1 crossover* (C1X) (Reeves, 1996) randomly selects a crossover point and generates child chromosomes taking the part before the crossover point of a parent and completing the chromosome by taking the legitimate elements in the order they appear in the other parent. In doing so, the C1 crossover preserves the absolute positions of the nurses in the part of the chromosome in front of the crossover points, while the relative positions are preserved in the part after the crossover point.

The *order-based crossover* (OBX) (Davis, 1985) randomly chooses two crossover points. The crossover operator copies the chromosome part between the crossover points of the two parents to the respective child chromosome while preserving the relative order of the nurses indicated by the other parent.

The *uniform order based crossover* (UOX) (Syswerda, 1996) randomly generates a crossover template of 0s and 1s. The 1s are copied from one of the parents whereas the positions of the 0s are filled in the order in which they appear in the other parent chromosome.

The *partially mapped crossover* (PMX*)* (Goldberg and Lingle, 1985) uses two crossover points to create the child solutions. The section between these two points defines an interchange mapping as illustrated in table 2.

The *parameterised uniform order crossover* (PUX) is a simple extension of the uniform order based crossover of Syswerda (1996). However, the creation of the binary vector is influenced by a parameter p (between 0 and 1) to measure the probability of a 0 and 1 in the vector.

Figure 3 illustrates the various crossover operators on the example NSP instance of figure 1. The nurse-data representation of the two parents is equal to the chromosomes [1 2 3 4 5] and [4 3 2 5 1]. These chromosomes represent the sequence in which nurses will be scheduled, based on the k-shortest path algorithm, which will lead to the parent schedules of figure 1. The left column specifies the crossover operator details, the middle column the parent chromosomes and the right column shows the newly created child solutions expressed in the nurse-data representation.

Crossover Operator	Parent Solutions	Constructed Child Solutions
C1X: Two child chromosomes are constructed based on the indicated assignments in both parent chromosomes which are chosen based on a randomly generated crossover point between positions 2 and 3. The indicated assignments are copied to the child chromosomes, while the remaining chromosome positions are constructed by preserving the relative positions of the nurses of the other parent.	**Parent Solution 1** 1 2 3 4 5 **Parent Solution 2** 4 3 2 5 1	**Child Solution 1** 1 2 4 3 5 **Child Solution 2** 4 3 1 2 5
OBX: Two child chromosomes are constructed based on the indicated assignments in both parent chromosomes which are chosen based on two randomly generated crossover points between positions 2 and 3 and between positions 4 and 5. The copy-paste to the child chromosomes is identical to the C1 operator.	**Parent Solution 1** 1 2 3 4 5 **Parent Solution 2** 4 3 2 5 1	**Child Solution 1** 2 5 3 4 1 **Child Solution 2** 1 3 2 5 4
UOX: Two child chromosomes are constructed based on the indicated assignments in both parent chromosomes which are chosen based on a randomly generated nurse vector 10101. The copy-paste to the child chromosomes is identical to the C1 operator.	**Parent Solution 1** 1 2 3 4 5 **Parent Solution 2** 4 3 2 5 1	**Child Solution 1** 1 4 3 2 5 **Child Solution 2** 4 3 2 5 1
PMX: Two child chromosomes are constructed based on the indicated assignments in both parent chromosomes which are chosen based on two randomly generated crossover point between positions 2 and 3 and positions 4 and 5. The chromosome positions between the two points are copied to the child chromosomes and define an interchange mapping. The remaining positions are copied from the other parent chromosome, followed by an interchange, as follows: 3 is replaced by 2 and 4 is replaced by 5 for the first child and vice versa for the second child solution.	**Parent Solution 1** 1 2 3 4 5 **Parent Solution 2** 4 3 2 5 1	**Child Solution 1** 5 2 3 4 1 **Child Solution 2** 1 3 2 5 4
PUX: (similar to the uniform crossover operator) Two child chromosomes are constructed based on the indicated assignments in both parent chromosomes which are chosen based on a generated nurse vector 10001 with each binary cell a certain probobility p = 0.40 to be equal to 1.	**Parent Solution 1** 1 2 3 4 5 **Parent Solution 2** 4 3 2 5 1	**Child Solution 1** 1 4 3 2 5 **Child Solution 2** 4 2 3 5 1

Figure 3. The crossover operators used in a nurse-pattern representation

3 Computational results

In this section, we present computational results of our genetic algorithm on a subset of the NSPLib problem instances of Vanhoucke and Maenhout (2005) to investigate the effectiveness of the various crossover operators. This dataset contains 4 sub-sets of problem instances with respectively 25 (N25 set), 50 (N50), 75 (N75), and 100 (N100) nurses and a 7-days scheduling horizon (this so-called *diverse set* contains 4 * 7290 instances). The dataset also consists of a *realistic set* with 2 sub-sets where respectively 30 (N30) and 60 (N60) nurses need to be scheduled over a period of 28 days. The nurses' preference structure and the coverage requirements of each sub-set are characterized by systematically varied levels of various NSP complexity indicators proposed in Vanhoucke and Maenhout (2005). All sets have been extended by 8 different combinations of case-specific constraints which appear frequently in literature (Cheang et al., 2003), i.e. the minimum/maximum number of working assignments, the minimum/maximum number of assignments per shift type, the minimum/maximum consecutive working assignments, and the minimum/maximum consecutive working assignments per shift type. The sets of problem instances as well as the 8 case-specific constraint files can be downloaded from http://www.projectmanagement.ugent.be/nsp.php. In the remainder of this section, we have tested the different crossover operators on the 25-nurses diverse sub-set (7 days), as well as on the 30-nurses realistic sub-set (28 days). The tests have been carried out on a Toshiba SPA10 with a 2.4 Ghz processor and 256 Mb RAM, under a stop criterion of 1,000 evaluated schedules. The population size depends on the size of the problem instances, and equals 30 (25) for the N25 (N30) instances. The mutation operator assigns randomly generated schedules to 20% of the nurses every 10 moves. The selection of parent solutions has been done in a complete random way. The specific settings of the local search approach are completely similar to the settings of Maenhout and Vanhoucke (2005b).

In the section 3.1, we give an overview of the computational performance of our algorithm applying the various crossover operators. In section 3.2, we investigate three different hybridizations of these different crossover operators. In section 3.3, we present best known solutions for our large dataset based on the best hybridization technique.

3.1 Comparison of the various crossover operators

The results are displayed in figures 4 (25 nurses) and 5 (30 nurses) averaged over the eight different versions of case-specific constraints. In order to display the quality of the schedule from both the individual nurses' point-of-view (expressed as the total preference cost) as well as from the total nurses' requirements point-of-view (expressed as the total penalty cost of coverage violations), we display the results in two dimensions. The horizontal axis measures the average cost of coverage violations using a unit penalty cost of 100. The vertical axis measures the total preference costs over all nurses. The diagonal dotted lines display all possible (x-axis, y-axis) combinations with a similar total cost (preference and coverage) and are used to compare the total performance of the various crossover operators.

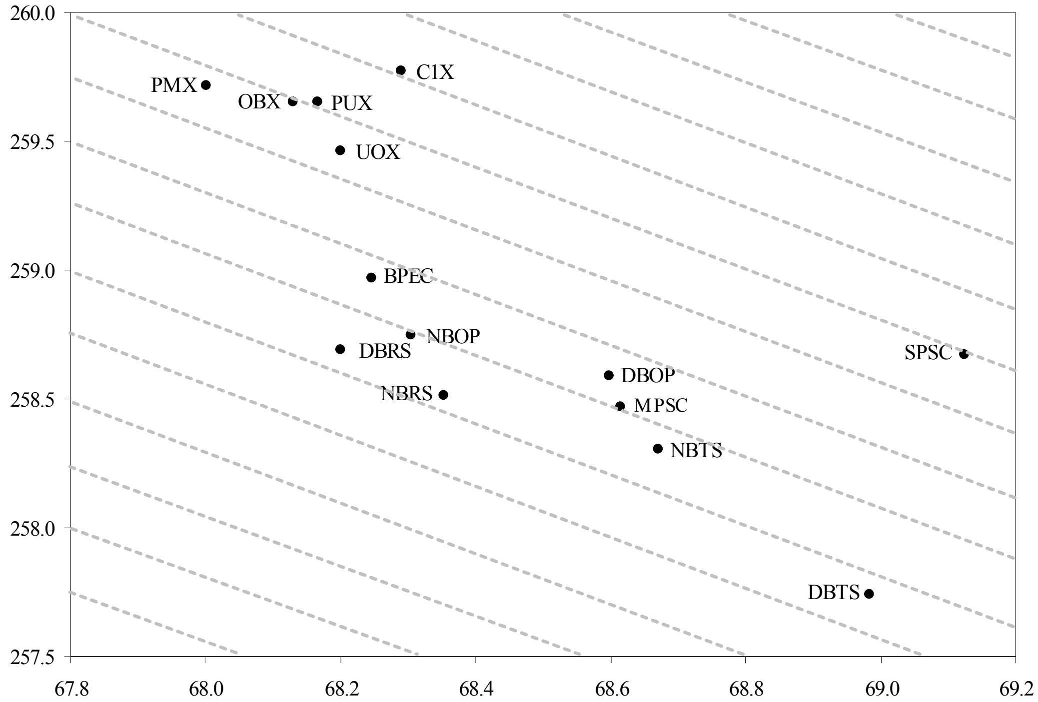

Figure 4. Computational performance of the different crossover operators (N25 instances)

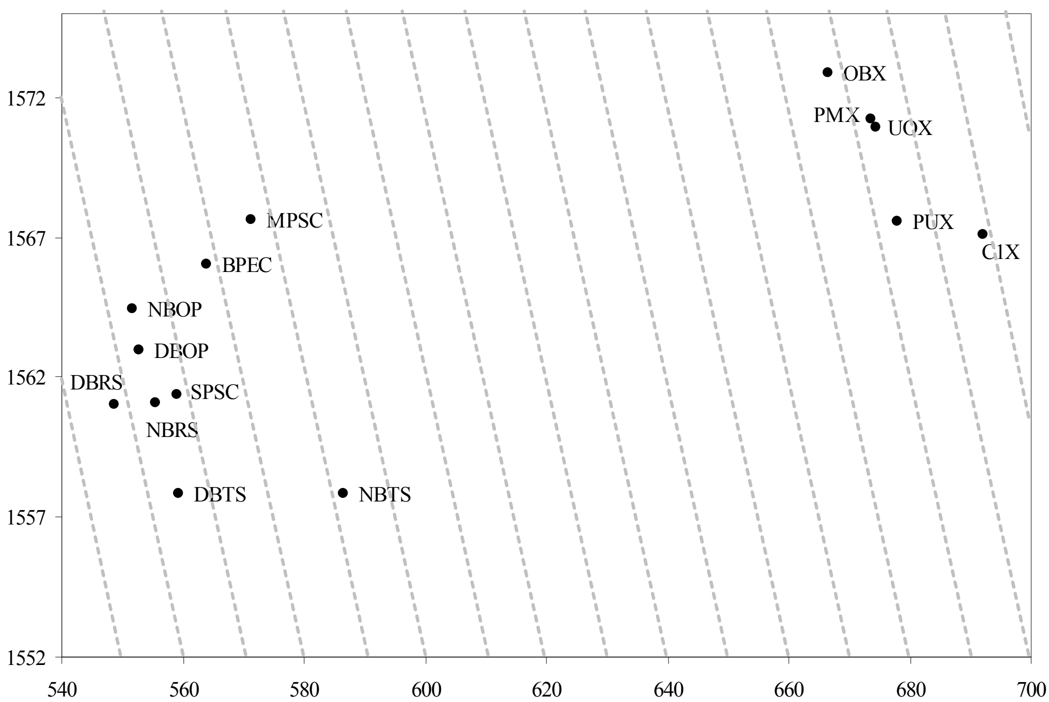

Figure 5. Computational performance of the different crossover operators (N30 instances)

The figures reveal that the nurse-day view crossover operators outperform the nurse-pattern view operators. For the N25 instances, the nurse-day operators clearly outperform the nurse-pattern view operators in terms of preference cost, but have a slightly worse performance in terms of coverage violations. However, the overall solution quality (preference and coverage) is the best for all nurse-day based crossover operators. The N30 graph shows that the nurse-day view operators outperform the nurse-pattern view operators, both in terms of preferences and coverage violations. In the remainder of this paper, we will rely on the nurse-day view crossover operators and not on the nurse-pattern view operators. To that purpose, we rely on both figures to select various nurse-pattern based crossover operators as potential candidates for hybridization.

3.2 A hybrid genetic algorithm

In this section, we use the computational results of previous section to construct a hybridized genetic algorithm based on the various crossover operators. More precisely, we combine the different operators under three different methods. First, the simple variable neighbourhood search applies each crossover operator in a specific proportion during the search. Second, the parent-related hybridization selects a specific crossover operator based on information of the parents during the search process. Last, an instance-specific hybridization selects a specific crossover based on a-priori calculations of different characteristics of the problem instance. These three hybridized versions have been tested on the N25 and N30 sub-sets of the NSPLib dataset for all sets of constraints.

3.2.1 Variable neighbourhood hybridization (VNH)

In this simple variable neighbourhood search, each crossover operator will be selected with a certain probability. Hence, this principle is characterized by a weighted average of a set of crossover operators which indicates the proportion each crossover operator is applied. We have implemented three versions of this approach, each time taking a sub-set of different crossover operators into account.

- VNH-all: in this generic hybridized approach, *all* crossover operators have an equal probability to be selected.
- VNH-best: in this hybridized approach, a combination of the overall three *best* operators is used. The three best performing crossover operators for the N25 instances have been selected from figure 4 and are DBTS, NBRS and DBRS. These three best single crossover operators lead to the lowest objective function value, in terms of preference costs and coverage penalty costs (see the diagonal lines). The best performing crossover operators (in terms of total objective function value) for the N30 instances are equal to DBRS, DBOP, NBOP (referred to as VNH-best1). However, since other crossover operators have a lower preference cost (although the total cost is higher than any of the three previous crossover operators), we also have tested two other combinations, i.e. NBRS, DBTS and NBTS (VNH-best2) and DBRS, DBTS and NBRS (VNH-best3). The relative importance of each crossover has been fine-tuned on a sub-set and equals, respectively, 50%, 40% and 10% (40%, 30%, and 30%) for the N25 (N30) sub-set.

- VNH-type: in this hybridized approach, the best combination of three crossover operators is used. Therefore, we have evaluated all combinations of three operators and selected the combination with the best performance over all problem instances. Moreover, these calculations lead to a combination where each crossover operator belongs to a *different type*, such that the combination is a mix of a nurse-based, day-based and specific crossover operator. The selected combinations are the NBRS, DBRS, and BPEC (NBRS, DBRS, and BPEC) crossover operators for the N25 (N30) sub-set. The relative importance of each crossover was fine-tuned and amounts respectively 45%, 40% and 15% (30%, 60%, and 10%) for the N25 (N30) sub-set.

3.2.2 Parent-related hybridization (PRH)

The parent-related hybridization makes use of pre-calculated information to determine which crossover operator to apply on the parent solutions. During the construction of a child solution, the algorithm chooses between a nurse-based, a day-based or a specific crossover operator (see section 2.2.1) based on information from the parents. To that purpose, the algorithm calculates three different values measuring the diversity of two parents related to the three types of crossover operators. These values will be used to explore the potential of each specific operator, as follows:

- The *nurse-based value* equals the absolute difference between the preferences of a nurse's schedule in both parent solutions divided by the nurse's minimal preference cost (thereby ignoring all case-specific constraints). The average over all nurses measures the potential of a nurse-based crossover operator.
- The *day-based value* measures the potential of the day-based crossover operators and is analogue to the nurse-based value. It is equal to the average absolute deviation between the preferences of each day for both parents divided by the minimal preferences for each day.
- The *specific-based value* measures the potential of the BPEC and is equal to the absolute difference between the total preference costs of the complete schedules of both parent solutions divided by the minimal preference cost (ignoring all case-specific constraints).

During the meta-heuristic search, the algorithm calculates the three values each time a child will be constructed and selects the crossover with the highest value to construct the child solution. Consequently, this hybridization mechanism acts as a diversification mechanism, since it selects a specific crossover based on the diversity of the two parents. In our approach, the algorithm selects between the NBRS, DBRS and BPEC (NBRS, DBRS and BPEC) operators for the nurse-based, day-based and specific crossover operator for the N25 (N30) instances, respectively. The three selected crossover operators is the best combination of three-operators as found by the VNH-type.

3.2.3 Instance-related hybridization (IRH)

Rather than selecting crossover operators during the search process based on information from both parents, we explore the ability of choosing one particular crossover operators throughout the whole search process depending on the characteristics of the problem instances. To that purpose, we rely on the systematically varied levels of the proposed complexity indicators of Vanhoucke and Maenhout (2005) under which the data instances have been generated. Indeed, each problem instance is characterized by a combination of six complexity indicators which measure the structure of the preference matrix and the corresponding coverage requirements. More precisely, the *preference indicators* measure the diversity of preferences among the nurses, the diversity of nurse's preferences among shifts and diversity of nurse's preferences among days. The *coverage indicators* measure the total number of required working shifts, the distribution of these coverage requirements among the days and the distribution of these coverage requirements among the shifts. For each combination of indicators, the best meta-heuristic procedure in terms of average solution quality can be determined and applied on the N25 and N30 sub-sets.

3.2.4 Computational results for VNH, PRH and IRH

Figures 6 and 7 display the average performance of the hybridized crossover operators in the same manner as figures 4 and 5. The figures compare the performance of the various hybridizations with the nurse-day view crossover operators and clearly illustrate the beneficial effect of combining the various crossover operators into one hybridized version. Three hybrid crossover operators outperform the best performing crossover operator (DBTS) for the N25 instances, while all hybrid operators outperform the best single crossover operators (DBRS) for the N30 instances, in terms of total costs (preference and coverage). In general, the three simple hybrid crossover operators (VNH) have been further improved by adding parent-related (PRH) or instance-specific (IRH) information. Hence, intrinsic information about the problem structure, measured in terms of six complexity indicators (see Vanhoucke and Maenhout (2005)) leads to the best performing results in terms of individual nurses' preferences as well as in satisfying the minimal coverage requirements. These complexity indicators should be used during the further development of meta-heuristic procedures, since they can open doors to build intelligent decision-support systems that selects problem-specific parameters based on a-priori calculations of the problem structure.

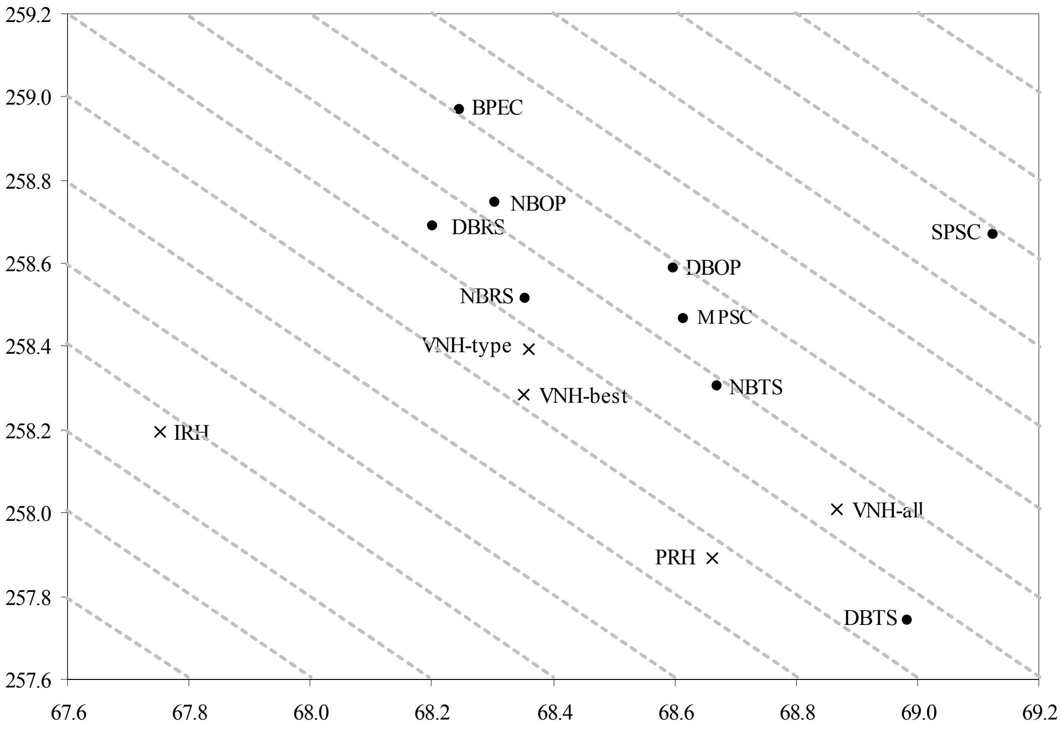

Figure 6. Computational performance for the hybridized crossover operators (N25 instances)

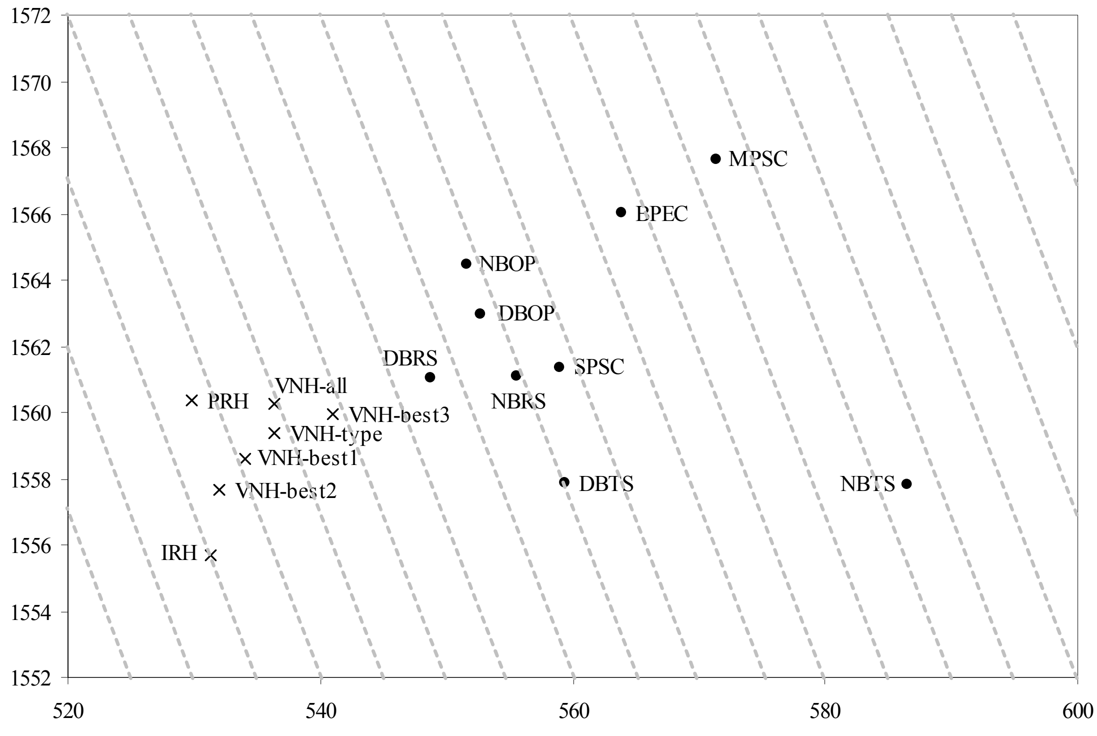

Figure 7. Computational performance for the hybridized crossover operators (N30 instances)

In order to illustrate the power of the intrinsic problem structure measured by the six indicators, we have tested our GA procedure on the N50, N75 and N100 instances (7 days horizon) and the N60 instances (28 days scheduling horizon) with the same information used for the N25 and N30 instances, respectively, under a stop criterion of 1,000 evaluated schedules. The results reveal that 28.85%, 28.54%, 33.03% and 40.42% of the solutions found outperform any previously found solution by the algorithms of Maenhout and Vanhoucke (2005a, 2005b) for the respective N25, N50, N75 and N100 sub-sets of the diverse set, whereas 25.77% and 23.57% of the previously found solutions are outperformed for the respective N30 and N60 sub-sets of the realistic set. These improved solutions can be downloaded from updated tables at www.projectmanagement.ugent.be/NSPLibSolutions.php

4 Conclusion and future research

In this paper, we have presented a hybrid genetic algorithm for the well-known nurse scheduling problem. In constructing a nurse roster over a pre-defined scheduling horizon, it is our objective to assign nurses to shifts in order to meet the minimal coverage constraints and other case-specific constraints and to satisfy the individual nurses' preferences as much as possible.

We have compared the data representation schemes and the crossover operators of various algorithms proposed in literature on a standard dataset under a strict test design with a strict stop criterion to facilitate comparison between procedures. We have hybridized different crossover operators by means of three techniques. First, we investigated a simple hybridization of the crossover operators by using a weighted average of certain crossover operators. Secondly, we have made use of pre-calculated parent-related information based on which we decide which crossover operator to use. Thirdly, we exploit information about the characteristics of the problem instance to determine which crossover operator to use. We have illustrated the contribution of hybrid algorithms over the use of single crossover operators. Exploring various neighbourhoods and hybrid algorithms lead to promising results and hence might have a bright future in the further development of meta-heuristic optimization algorithms.

Our future research avenues will focus on the evaluation and comparison of various optimization principles for the nurse scheduling problem. Further analysis of the potential of hybridization based on information during the search process (e.g. parent-related information) or a-priori calculations of instance parameters (e.g. instance-related information) will allow us to set-up an intelligent decision-support system to facilitate and improve the selection of optimization procedures. We believe that we have clearly illustrated the potential of hybridization for the nurse scheduling problem.

References

Aickelin, U., and Dowsland, K.A., 2000, "Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem", *Journal of Scheduling*, 3, 139-153.

Aickelin, U., and Dowsland, K.A., 2004, "An indirect Genetic Algorithm for a nurse-scheduling problem", *Computers and Operations Research*, 31, 761-778.

Burke, E.K., Cowling, P., De Causmacker, P. and Vanden Berghe, G., 2001, "A memetic approach to the nurse rostering problem", *Applied Intelligence*, 3, 199-214.

Burke, E.K., De Causmaecker, P., Vanden Berghe, G. and Van Landeghem, H., 2004, "The state of the art of nurse rostering", *Journal of Scheduling*, 7, 441-499.

Cheang, B., Li, H., Lim, A., and Rodrigues, B., 2003, "Nurse rostering problems – a bibliographic survey", *European Journal of Operational Research*, 151, 447-460.

Dias, T.M., Ferber, D.F., de Souza, C.C., Moura, A.V., 2003, "Constructing nurse schedules at large hospitals", *International Transactions in Operational Research*, 10, 245-265.

Davis, L., 1985, "Applying adaptive algorithms to epistatic domains", *Proceedings of the Ninth International Joint Reference on Artificial Intelligence*, 1, 162-164.

Ernst, A.T., Jiang, H., Krishamoorty, M., Owens, B., and Sier, D., 2004a, "Staff scheduling and rostering: A review of applications, methods and models", *European Journal of Operational Research*, 153, 3-27.

Ernst, A.T., Jiang, H., Krishamoorty, M., Owens, B., and Sier, D., 2004b, "An Annotated Bibliography of Personnel Scheduling and Rostering", *Annals of Operations Research*, 127, 21-144.

Felici, G., and Gentile, C., 2004, "A polyhedral Approach for the Staff Rostering Problem", *Management Science*, 50, 381-393.

Glover, F., and McMillan, C., 1986, "The General Employee Scheduling Problem: An integration of MS and AI", *Computers and Operations Research*, 13, 563-573.

Goldberg, D.E., and Lingle, R., "Alleles, loci, and the travelling salesman problem", In: Grefenstette, J. (editor), *Proceedings of the First International Reference on Genetic Algorithms and their Applications*, Hilsdale, NJ: Lawrence Erlbaum Associates Publishers, 1985, 154-159.

Hansen, P., and Mladenovic, N., 2001, "Variable neighborhood search: Principle and applications", *European Journal of Operational Research*, 130, 449-467.

Holland, J.H., 1975, "Adaptation in Natural and Artificial Systems", *University of Michigan Press*, Republished by MIT Press (1992).

Inoue, T., Furuhashi, T., 2003, "A Proposal of Combined Method of Evolutionary Algorithm and Heuristics for Nurse Scheduling Support System", *IEEE Transactions on Industrial Electronics*, 50, 833-841.

Jan, A., Yamamoto M., and Ohuchi, A., 2000, "Evolutionary Algorithms For Nurse Scheduling Problem", P*roceedings of the 2000 Congress on Evolutionary Computation*, 196-203.

Kapsalis, A., Rayward-Smith, V.J., and Smith, G.D., 1993, "Solving the graphical Steiner tree problem using genetic algorithms", *Journal of the Operational Research Society*, 44,397-406.

Michalewicz, Z., 1995, "A survey of constraint handling techniques in evolutionary computation methods", *Proceeding of the Fourth Annual Conference on Evolutionary Programming*, San Diego, California, 135-155.

Maenhout, B. and Vanhoucke, M., 2005a, "An Electromagnetism meta-heuristic for the nurse scheduling problem", *working paper 05/316*, Ghent University.

Maenhout, B. and Vanhoucke, M., 2005b, "New Computational Results for the Nurse Scheduling Problem: A Scatter Search Algorithm", *working paper 05/341*, Ghent University.

Osogami, T., and Imai, H., 2000, "Classification of Various Neighbourhood Operations for the Nurse Scheduling Problem", *Lecture Notes in Computer Science*, 1969, 72-83.

Reeves, C.R., 1995, "A Genetic Algorithm for Flowshop Sequencing", *Computers and Operations Research*, 22, 5-13.

Reeves, C.R., 1996, "Hybrid genetic algorithms for bin-packing and related problems", *Annals of Operations Research*, 63, 371-396.

Syswerda, G., 1996, "Schedule optimisation using genetic algorithms", In: Davis, L., editor, *Handbook of genetic algorithms*, London: International Thomson Computer Press, 1996, 335-349.

Thompson, G.M., 1995, "Improved implicit optimal modelling of the labour shift scheduling problem", *Management Science*, 43, 595-607.

Vanhoucke, M., and Maenhout, B., 2005, "Characterisation and Generation of Nurse Scheduling Problem Instances", *working paper 05/339*, Ghent University.

Warner, H.W., 1976, "Scheduling Nursing Personnel According to Nursing Preference: A Mathematical Approach", *Operations Research*, 24, 842-856.