5. Analysis and discussion

In this section, the effect of parameters initial value of \( \text{Swap\_Selection\_Probability} \), \( \text{Density} \), \( \text{Stagnation\_Limit} \), \( K \) and \( M \) to algorithms’ performance is investigated and the results of different specific parameter values are demonstrated. Due to the fact that there are no obvious criteria for defining specific parameter values of the proposed algorithm for all instances of the problem, we have selected these values by trial and error. More precisely, we have conducted exhaustive experiments and selected the values that achieved the best simulation results.

5.1 Investigating the effect of initial value of \( \text{Swap\_Selection\_Probability} \) to algorithms’ performance

In Fig. 8, experimental results that investigate the effect of the initial value of \( \text{Swap\_Selection\_Probability} \) to algorithms’ performance are presented for six INRC-2010 instances.
A two-phase adaptive variable neighborhood approach for nurse rostering

![Graph of Sprint07](image1)

![Graph of Sprint_late01](image2)

![Graph of Sprint_hidden07](image3)
As shown in Fig. 8, the lowest average fitness values, for all these instances, are achieved by setting the initial value of \( \text{Swap\_Selection\_Probability} \) equal to 0.3. Since the same holds for the majority of INRC-2010 instances, we decided to set the initial value of \( \text{Swap\_Selection\_Probability} \) equal to 0.3 for all experiments conducted.

5.2 Investigating the effect of parameter Density to algorithms’ performance

The average value of parameter \( \text{Density} \), over all INRC-2010 instances, is approximately 0.55 while its minimum and maximum value is 0.45 and 0.7, respectively. The instance with the minimum \( \text{Density} \) value is long_hidden05 and belongs to the Long track. Instances medium01, medium02, medium03, medium04 and medium05, which have the maximum \( \text{Density} \) value, belong to the Medium track. The \( \text{Density} \) value of all instances belonging to the Long track lies in \([0.45, 0.54]\). Although there are other instances with bigger \( \text{Density} \) values, instances belonging to the Long track are characterized as the most difficult ones; hence the organizers of the competition have allowed the most computational time for them to be solved. We observed that the proposed algorithm solves easily instances having a big \( \text{Density} \) value, such as medium01, medium02 and medium05 (\( \text{Density}=0.7 \)), while it fails at instances, such as sprint_hidden06, sprint_hidden07 and sprint_hidden08 with density values equal to 0.5, 0.48 and 0.54, respectively. This observation demonstrates that the \( \text{Density} \) value does not constitute an obvious metric of an instance’s difficulty.

Nevertheless, we came to this conclusion after having curried out extensive experiments without being able to predict a priori such a situation. That is the reason why we have employed parameter \( \text{Density} \) in the formula that computes the \( \text{Swap\_Selection\_Probability} \) value, since we had no other indication about each instance’s difficulty. As seen in Fig. 6,
according to the formula used for the determination of the \textit{Swap\_Selection\_Probability} value, the higher the \textit{Density} value is the greater the \textit{Swap\_Selection\_Probability} becomes. Hopefully, by following such a practice, the proposed algorithm manages to achieve very satisfactory results, as reported in Section 4.

5.3 \textit{Investigating the effect of parameters K, M and Density to Swap\_Selection\_Probability’s value}

In Fig. 9, 10 and 11, we investigate the effect of parameters \textit{K}, \textit{M} and \textit{Density} to \textit{Swap\_Selection\_Probability}’s value. The equation used to estimate the \textit{Swap\_Selection\_Probability}’s value is the following (Fig. 6):

\[
\text{Swap\_Selection\_Probability} = 1 + 0.4 \cdot \text{Density} - K \cdot \frac{1 + 0.8 \cdot \text{Density}}{K + M}
\]

where \textit{K}, \textit{M} are positive integers greater than zero. If the value estimated by the above equation is smaller than 0 the \textit{Swap\_Selection\_Probability}’s value is set to 0. If the value estimated by the above equation is bigger than 1 the \textit{Swap\_Selection\_Probability}’s value is set to 1. However, in real world problems, these cases are extremely rare according to all experiments conducted.

In Figure 9, we set parameter \textit{Density} equal to its average value (0.55), we set parameter \textit{K} equal to 1, 5, 10, 20, 50, 100 and 200 and we investigate how parameter \textit{M} affects the value of \textit{Swap\_Selection\_Probability}. As seen in Fig.9, if we keep parameters \textit{K} and \textit{Density} constant, the value of \textit{Swap\_Selection\_Probability} is getting bigger as parameter \textit{M} increases. This means that the more procedure \textit{Swap\_Random\_Rosters}() improves the global best, the bigger the value of \textit{Swap\_Selection\_Probability} gets. As a result, the algorithm is oriented to select the other swap procedure, that is, \textit{Swap\_Ordered\_Rosters}(). This sustains a balanced variable neighborhood search between these two neighborhoods.

In Figure 10, we set parameter \textit{Density} equal to its average value (0.55), we set parameter \textit{M} equal to 1, 5, 10, 20, 50, 100 and 200 and we investigate how parameter \textit{K} affects the value of \textit{Swap\_Selection\_Probability}. As seen in Fig.10, if we keep parameters \textit{M} and \textit{Density} constant, the value of \textit{Swap\_Selection\_Probability} is getting smaller as parameter \textit{K} increases. This means that the more procedure \textit{Swap\_Ordered\_Rosters}() improves the global best, the smaller the value of \textit{Swap\_Selection\_Probability} gets. As a result, the algorithm is oriented to select the other swap procedure, that is, \textit{Swap\_Random\_Rosters}(). This sustains a balanced variable neighborhood search between these two neighborhoods.
In Figure 11 and Figure 12, we investigate how parameter Density affects the value of Swap_Selection_Probability. In Figure 11, parameters $K$ and $M$ are set to thirty different pairs of values where $K<M$. In Figure 12, parameters $K$ and $M$ are set to thirty different pairs of values where $K>M$. We omitted deliberately cases where $K=M$, since in these cases the Swap_Selection_Probability is always 0.5.

As seen in Fig.11, if we keep parameters $K$ and $M$ constant ($K<M$), the value of Swap_Selection_Probability is getting bigger as parameter Density increases. This means that, in case Swap_Random_Rosters() has improved the global best more times than Swap_Ordered_Rosters(), the more dense an instance is, the bigger the value of
Swap_Selection_Probability gets. As a result, the algorithm, for more dense instances, is oriented to select the other swap procedure, that is, Swap_Ordered_Rosters(). This sustains a balanced variable neighborhood search between these two neighborhoods for more dense instances.

As seen in Fig.12, if we keep parameters $K$ and $M$ constant ($K>M$), the value of Swap_Selection_Probability is getting smaller as parameter Density increases. This means that, in case Swap_Ordered_Rosters() has improved the global best more times than Swap_Random_Rosters(), the more dense an instance is, the smaller the value of Swap_Selection_Probability gets. As a result, the algorithm, for more dense instances, is oriented to select the other swap procedure, that is, Swap_Random_Rosters(). This sustains a balanced variable neighborhood search between these two neighborhoods for more dense instances.
5.4 Investigating the convergence behaviour of the algorithm in comparison to the evolution of $K$, $M$ and Swap Selection Probability parameters’ values

In Fig. 13, experimental results that investigate the convergence behaviour of the algorithm in comparison to the evolution of $K$, $M$ and Swap Selection Probability parameters’ values is presented for three INRC-2010 instances, namely, sprint_late03, sprint_late04 and sprint_late07.

---

**Fig. 12.** Investigating the effect of parameter Density to Swap Selection Probability’s value when $K>M$
Fig. 13. Investigating the convergence behaviour of the algorithm in comparison to the evolution of \( K \), \( M \) and \( \text{Swap\_Selection\_Probability} \) parameters’ values

As seen in Fig. 13, whenever parameter \( K \) is increased parameter \( M \) is stable and vice versa. This is expected, since procedures \( \text{Swap\_Ordered\_Rosters}() \) and \( \text{Swap\_Random\_Rosters}() \) cannot be both applied during the same generation. In most cases, if the application of one of these procedures improves global best, then the algorithm continues to select the same swap procedure. If however the successive application of one of these procedures does not improve global best, the algorithm decides to apply the other swap procedure. This sustains a balanced variable neighborhood search between these two neighborhoods, especially in cases where the algorithm cannot improve the global best for a number of generations.

Regarding \( \text{Swap\_Selection\_Probability} \) and fitness function value evolution, one can observe the following. There are cases where an improvement in the global fitness value is followed
by a change in the \textit{Swap\_Selection\_Probability} (this is a case where the application of \textit{Swap\_Ordered\_Rosters}() or \textit{Swap\_Random\_Rosters}() improves global best). Also, there are other cases where an improvement in the global fitness value is not followed by a change in the \textit{Swap\_Selection\_Probability} (this is a case where the application of \textit{Swap\_1}(), \textit{Swap\_2}(), \textit{Swap\_3}(), \textit{Swap\_4}(), \textit{Swap\_5}(), \textit{Swap\_6}() and \textit{Swap\_Two\_Rosters}() improves global best). Finally, there are cases where both \textit{Swap\_Selection\_Probability} and fitness function value are stable (this is a case where no swap procedure manages to improve global best).