

НТУУ «КПИ»
ИНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО
АНАЛИЗА
КАФЕДРА СИСТЕМНОГО ПРОЕКТИРОВАНИЯ

Презентация к дипломной работе на тему: “Исследования алгоритмов оптимизации индивидуальных календарных планов обучения”



Выполнил: Севидов П.М.
Руководитель: Киселев Г.Д.

Киев 2016

Актуальность

С проблемой формирования календарного плана очень часто сталкиваются студенты. Разнообразие учебных предметов и работ приводит к неоптимальному использованию собственного времени и решений, учитывающих индивидуальные потребности, нету.

Цели

- ✓ Проанализировать алгоритмы оптимизации
- ✓ Реализовать выбранный алгоритм
- ✓ Сделать анализ результатов работы
- ✓ Сравнить результаты работы с работой алгоритма иммитации отжига

Вступ

- ✓ Сложности при формировании расписания появляются тогда, когда работ становится много, нужно учесть множество дополнительных условий и составить расписание не для одного человека, а для целого коллектива, когда существует сотни работ и исполнителей.
- ✓ Данный тип задач является NP сложной и не существует точной математической модели для нахождения оптимального решения. Поэтому целесообразно использование стохастических и эвристических методов.

Ограничивающие критерии

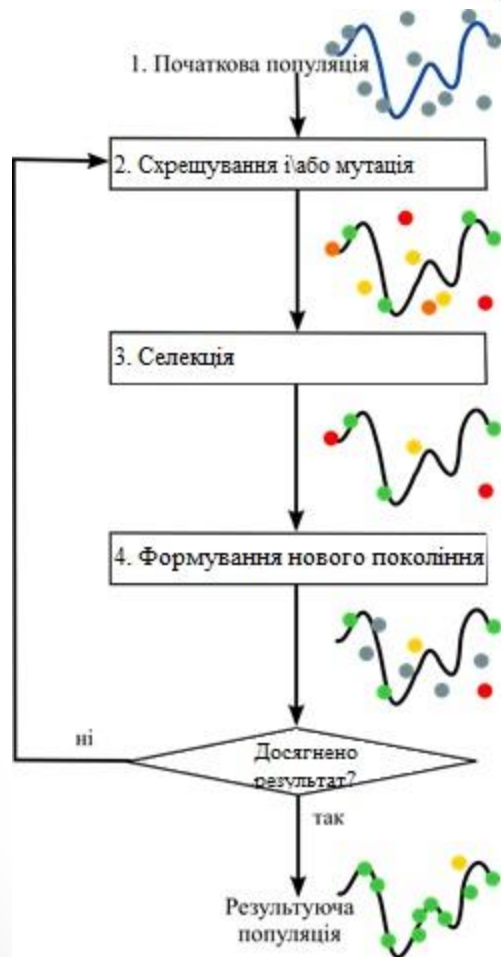
Hard Stop критерии (запрещено нарушать для данной задачи)

1. Каждая работа должна быть закончена и сдана до крайнего срока (deadline).
2. Прделанной работе можно сдать только в тот день, когда есть пара с преподавателем, который принимает работу.
3. Начать выполнение работы возможно при условии, если нет других дел в это время.

Soft Stop критерии (не желательно нарушать, нарушение приводит к ухудшению результата)

1. Обязательные задачи с высоким приоритетом должны выполняться как можно раньше.
2. Равномерность загруженности каждого рабочего дня (показатель загруженности не должен превышать допустимую норму).
3. Сумма дат завершения работ минимальна. Это условие отражает желание студента выполнить все работы как можно скорее.

Генетичний алгоритм



- ✓ Упорядоченний кроссинговер (OX1): часть одного из родителей берется за основу потомству. Далее в полученный потомок недостающие гены берутся из второго родителя сохраняя порядок и пропуская те, что уже есть в части от первого отца. Таким образом не нарушаются ни одно из ограничений.
- ✓ Лучшими показателями простоты реализации и быстродействия соответствует метод мутации обмена, меняет в последовательности двух соседей одного случайно выбранного гена.

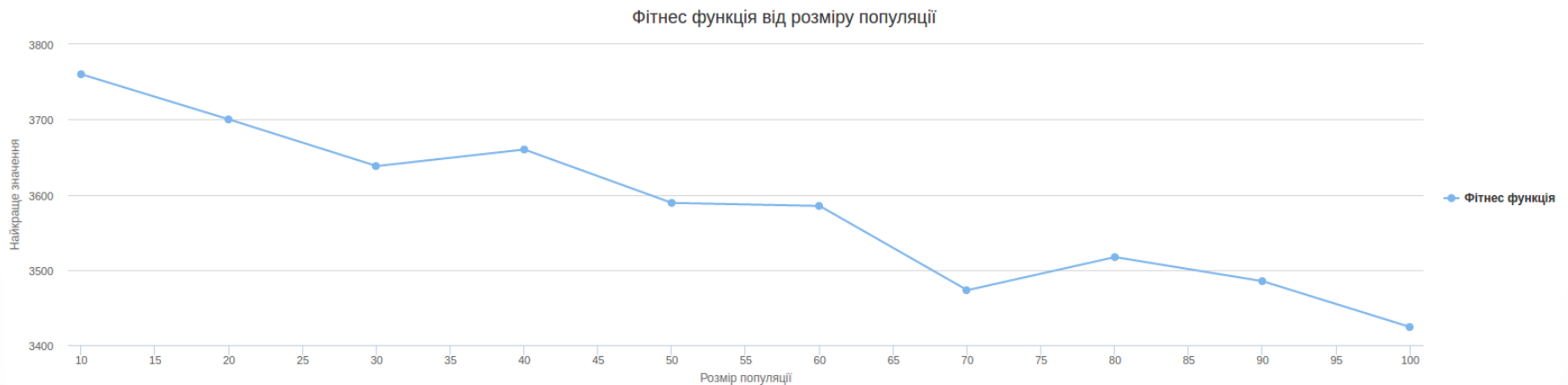
Фитнесс функция

$$F(\text{Schedule}_i) = \sum_{i=0}^n C_i * (w_i) * \frac{\sum_{k=0}^s \frac{\text{freeTime}_k - \text{sum}_k}{\text{freeTime}_k}}{s}$$

- 1) s – количество рабочих дней в семестре
- 2) k – номер рабочего дня
- 3) $\text{sum}_k = \sum_{i=0}^n p_{ik}$ – сумма продолжительности каждой работы, которая запланирована на k -й день

Зависимость наилучшего значения фитнес функции от размера популяции

При увеличении популяции, увеличивается вариативность особей, в свою очередь увеличивает шансы достичь оптимального результата. Минусом увеличения популяции является замедление работы алгоритма, поэтому нужно найти компромиссное решение. Как видно из рисунка таким значением является размер популяции в пределах 70



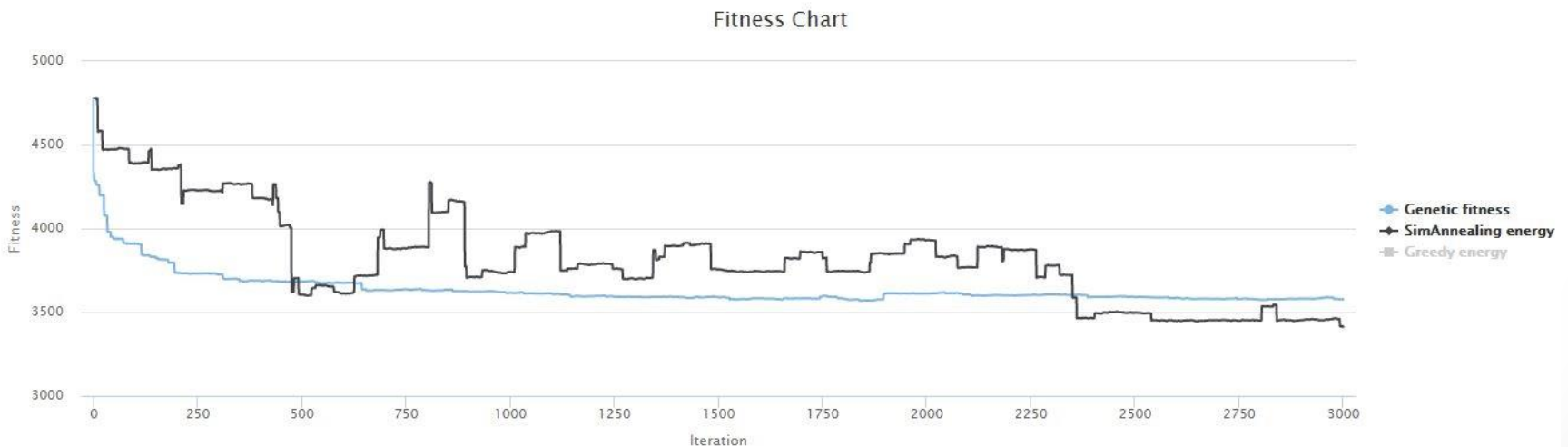
Зависимость наилучшего значения фитнес функции по популяции к вероятности мутации

При увеличении вероятности, возрастает «случайность» генерации значений нового поколения. Результатом данных исследований является выбор значения вероятности мутации в пределах от 0.1 до 0.25

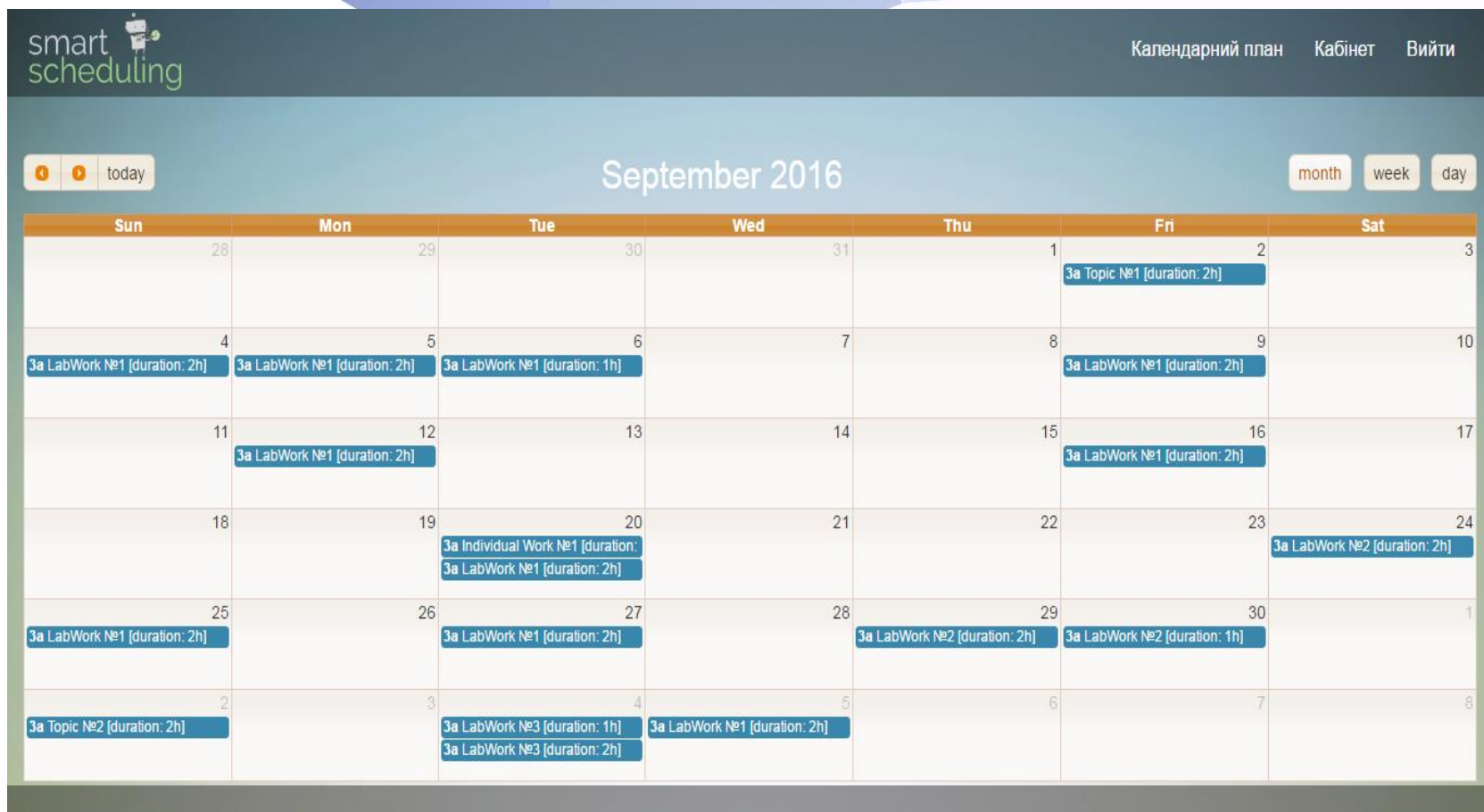


Сравнение оптимизации

- ✓ Генетический алгоритм в результате своей работы минимизировал значение функции на 26%
- ✓ Алгоритм имитации отжига в результате своей работы минимизировал значение функции на 29%



Сформированное расписание



Выводы

- ✓ Решена проблема построения индивидуального план-графика
- ✓ Был выбран упорядоченный кроссинговер (OX1) и метод мутации обмен
- ✓ Генетический алгоритм в результате своей работы минимизировал значение функции на 26%
- ✓ Финальная оптимизация генетического алгоритма меньше на 3%
- ✓ Можно достичь лучшую минимизацию увеличивая размер популяции, но возникает другая проблема - задействование невероятного количества памяти. Это приводит к частым запусков сборщика мусора, в результате чего проседания в скорости выдачи готового результата конечному пользователю.
- ✓ Недостатки алгоритма: Время исполнения функции оценки велико; Необходимо найти все решения задачи, а не одно из них; Много временных данных; Недоказанность сходимости;
- ✓ В дальнейшем возможно улучшить результаты, используя гибридные алгоритмы



Благодарю за
внимание!