

Biased Random Key Genetic Algorithm con Búsqueda Local para el Team Orienteering Problem

Alejandro Lix Klett

Directora: Prof. Dra. Irene Loiseau

Departamento de Computación

May 29, 2018

- 1 Orienteering Problem
- 2 Team Orienteering Problem
- 3 Ejemplo de solución de TOP
- 4 Metaheurísticas
- 5 Algoritmos Genéticos (GA)
- 6 Random Key Genetic Algorithm (RKGA)
- 7 Biased Random Key Genetic Algorithm (RKGA)
- 8 Second Section

Orienteering Problem

- Orientación es un deporte originario de Escandinavia

Orienteering Problem

- Orientación es un deporte originario de Escandinavia
- Cada jugador comienza en un punto de control y debe visitar tantos otros puntos de control como le sea posible dentro de un tiempo limite preespecificado.

Orienteering Problem

- Orientación es un deporte originario de Escandinavia
- Cada jugador comienza en un punto de control y debe visitar tantos otros puntos de control como le sea posible dentro de un tiempo limite preespecificado.
- Cada punto de control tiene un puntaje.

Orienteering Problem

- Orientación es un deporte originario de Escandinavia
- Cada jugador comienza en un punto de control y debe visitar tantos otros puntos de control como le sea posible dentro de un tiempo limite preespecificado.
- Cada punto de control tiene un puntaje.
- Cada punto de control puede ser visitado una sola vez a lo sumo.

Orienteering Problem

- Orientación es un deporte originario de Escandinavia
- Cada jugador comienza en un punto de control y debe visitar tantos otros puntos de control como le sea posible dentro de un tiempo limite preespecificado.
- Cada punto de control tiene un puntaje.
- Cada punto de control puede ser visitado una sola vez a lo sumo.
- El objetivo es maximizar el puntaje total.

Orienteering Problem

- Orientación es un deporte originario de Escandinavia
- Cada jugador comienza en un punto de control y debe visitar tantos otros puntos de control como le sea posible dentro de un tiempo limite preespecificado.
- Cada punto de control tiene un puntaje.
- Cada punto de control puede ser visitado una sola vez a lo sumo.
- El objetivo es maximizar el puntaje total.
- Este problema se conoce como Orienteering Problem (OP). El OP es NP-Hard como demostraron Golden, Levy y Vohra.

Team Orienteering Problem

- Hay M clientes, cada uno tiene un beneficio b_i y una coordenada en el plano.

Team Orienteering Problem

- Hay M clientes, cada uno tiene un beneficio b_i y una coordenada en el plano.
- Los puntos de salida y llegada tienen beneficio cero

Team Orienteering Problem

- Hay M clientes, cada uno tiene un beneficio b_i y una coordenada en el plano.
- Los puntos de salida y llegada tienen beneficio cero
- Hay N vehículos

Team Orienteering Problem

- Hay M clientes, cada uno tiene un beneficio b_i y una coordenada en el plano.
- Los puntos de salida y llegada tienen beneficio cero
- Hay N vehículos
- El beneficio de los clientes solo puede ser recolectado una vez.

Team Orienteering Problem

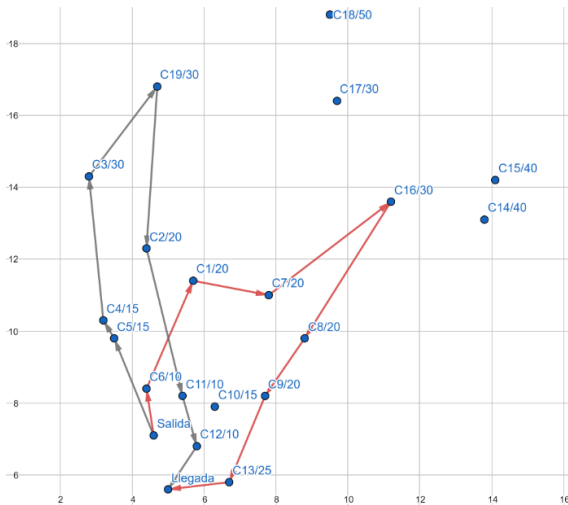
- Hay M clientes, cada uno tiene un beneficio b_i y una coordenada en el plano.
- Los puntos de salida y llegada tienen beneficio cero
- Hay N vehículos
- El beneficio de los clientes solo puede ser recolectado una vez.
- El objetivo es maximizar la sumatoria de los beneficios recolectados de todos los vehículos.

Team Orienteering Problem

- Hay M clientes, cada uno tiene un beneficio b_i y una coordenada en el plano.
- Los puntos de salida y llegada tienen beneficio cero
- Hay N vehículos
- El beneficio de los clientes solo puede ser recolectado una vez.
- El objetivo es maximizar la sumatoria de los beneficios recolectados de todos los vehículos.
- Como TOP contiene a OP, es al menos tan difícil.

Instancia p2.2.k del benchmark de Tsiligrides

La instancia tiene dos vehículos con un $d_{max} = 22,50$. Hay 19 clientes además de los puntos de salida y llegada.



Metaheurísticas

- Son métodos diseñados para encontrar buenas soluciones, en un tiempo razonable, a problemas de optimización combinatoria en general.

Metaheurísticas

- Son métodos diseñados para encontrar buenas soluciones, en un tiempo razonable, a problemas de optimización combinatoria en general.
- Las metaheurísticas son estrategias de alto nivel que guían una heurística específica del problema a resolver para mejorar su performance.

Metaheurísticas

- Son métodos diseñados para encontrar buenas soluciones, en un tiempo razonable, a problemas de optimización combinatoria en general.
- Las metaheurísticas son estrategias de alto nivel que guían una heurística específica del problema a resolver para mejorar su performance.

Características:

- Son estrategias que guían procesos de búsqueda.

Metaheurísticas

- Son métodos diseñados para encontrar buenas soluciones, en un tiempo razonable, a problemas de optimización combinatoria en general.
- Las metaheurísticas son estrategias de alto nivel que guían una heurística específica del problema a resolver para mejorar su performance.

Características:

- Son estrategias que guían procesos de búsqueda.
- Sus conceptos se pueden describir con un gran nivel de abstracción. No son para un problema específico.

Metaheurísticas

- Son métodos diseñados para encontrar buenas soluciones, en un tiempo razonable, a problemas de optimización combinatoria en general.
- Las metaheurísticas son estrategias de alto nivel que guían una heurística específica del problema a resolver para mejorar su performance.

Características:

- Son estrategias que guían procesos de búsqueda.
- Sus conceptos se pueden describir con un gran nivel de abstracción. No son para un problema específico.
- En muchos casos son algoritmos no-determinísticos.

Metaheurísticas

- Son métodos diseñados para encontrar buenas soluciones, en un tiempo razonable, a problemas de optimización combinatoria en general.
- Las metaheurísticas son estrategias de alto nivel que guían una heurística específica del problema a resolver para mejorar su performance.

Características:

- Son estrategias que guían procesos de búsqueda.
- Sus conceptos se pueden describir con un gran nivel de abstracción. No son para un problema específico.
- En muchos casos son algoritmos no-determinísticos.
- Sus desarrollos y diseños suelen estar motivados por comportamientos naturales.

Metaheurísticas

- Son métodos diseñados para encontrar buenas soluciones, en un tiempo razonable, a problemas de optimización combinatoria en general.
- Las metaheurísticas son estrategias de alto nivel que guían una heurística específica del problema a resolver para mejorar su performance.

Características:

- Son estrategias que guían procesos de búsqueda.
- Sus conceptos se pueden describir con un gran nivel de abstracción. No son para un problema específico.
- En muchos casos son algoritmos no-determinísticos.
- Sus desarrollos y diseños suelen estar motivados por comportamientos naturales.
- No garantizan que una solución óptima sea encontrada.

Metaheurísticas

- Son métodos diseñados para encontrar buenas soluciones, en un tiempo razonable, a problemas de optimización combinatoria en general.
- Las metaheurísticas son estrategias de alto nivel que guían una heurística específica del problema a resolver para mejorar su performance.

Características:

- Son estrategias que guían procesos de búsqueda.
- Sus conceptos se pueden describir con un gran nivel de abstracción. No son para un problema específico.
- En muchos casos son algoritmos no-determinísticos.
- Sus desarrollos y diseños suelen estar motivados por comportamientos naturales.
- No garantizan que una solución óptima sea encontrada.
- Las técnicas metaheurísticas van desde algoritmos simples de búsqueda local a complejos procesos de aprendizaje.

Algunas técnicas:

- Simulated Annealing

Algunas técnicas:

- Simulated Annealing
- Tabu Search

Algunas técnicas:

- Simulated Annealing
- Tabu Search
- Algoritmos evolutivos

Algunas técnicas:

- Simulated Annealing
- Tabu Search
- Algoritmos evolutivos
- Colonia de hormigas

Algunas técnicas:

- Simulated Annealing
- Tabu Search
- Algoritmos evolutivos
- Colonia de hormigas
- Variable Neighborhood Search

Algunas técnicas:

- Simulated Annealing
- Tabu Search
- Algoritmos evolutivos
- Colonia de hormigas
- Variable Neighborhood Search
- Iterated Local Search

Algunas técnicas:

- Simulated Annealing
- Tabu Search
- Algoritmos evolutivos
- Colonia de hormigas
- Variable Neighborhood Search
- Iterated Local Search
- Etc

- Motivados en el concepto de supervivencia del más apto.

Algoritmos Genéticos (GA)

- Motivados en el concepto de supervivencia del más apto.
- Los algoritmos genéticos manejan un conjunto de individuos.

Algoritmos Genéticos (GA)

- Motivados en el concepto de supervivencia del más apto.
- Los algoritmos genéticos manejan un conjunto de individuos.
- Cada individuo es un cromosoma que codifica una solución.

Algoritmos Genéticos (GA)

- Motivados en el concepto de supervivencia del más apto.
- Los algoritmos genéticos manejan un conjunto de individuos.
- Cada individuo es un cromosoma que codifica una solución.
- Cada cromosoma tienen asociado un nivel de condición física que está correlacionado con el correspondiente valor de la función objetivo de la solución que codifica.

Algoritmos Genéticos (GA)

- Motivados en el concepto de supervivencia del más apto.
- Los algoritmos genéticos manejan un conjunto de individuos.
- Cada individuo es un cromosoma que codifica una solución.
- Cada cromosoma tienen asociado un nivel de condición física que está correlacionado con el correspondiente valor de la función objetivo de la solución que codifica.
- En cada generación se crea una nueva población con individuos provenientes de tres fuentes distintas: crossover, elites y mutantes.

Random Key Genetic Algorithm (RKGA)

- Los individuos son representados por un vector de números reales en el intervalo $[0, 1]$.

Random Key Genetic Algorithm (RKGA)

- Los individuos son representados por un vector de números reales en el intervalo $[0, 1]$.
- La población inicial es generada al azar.

Random Key Genetic Algorithm (RKGA)

- Los individuos son representados por un vector de números reales en el intervalo $[0, 1]$.
- La población inicial es generada al azar.
- El decodificador es el responsable de convertir un cromosoma en una solución válida del problema.

Random Key Genetic Algorithm (RKGA)

- Los individuos son representados por un vector de números reales en el intervalo $[0, 1]$.
- La población inicial es generada al azar.
- El decodificador es el responsable de convertir un cromosoma en una solución válida del problema.
- En cada iteración se toman los mejores individuos y pasan directamente a la siguiente generación (elites).

Random Key Genetic Algorithm (RKGA)

- Los individuos son representados por un vector de números reales en el intervalo $[0, 1]$.
- La población inicial es generada al azar.
- El decodificador es el responsable de convertir un cromosoma en una solución válida del problema.
- En cada iteración se toman los mejores individuos y pasan directamente a la siguiente generación (elites).
- La mayoría de los individuos de la nueva generación se genera cruzando dos individuos de la generación actual (crossover).

Random Key Genetic Algorithm (RKGA)

- Los individuos son representados por un vector de números reales en el intervalo $[0, 1]$.
- La población inicial es generada al azar.
- El decodificador es el responsable de convertir un cromosoma en una solución válida del problema.
- En cada iteración se toman los mejores individuos y pasan directamente a la siguiente generación (elites).
- La mayoría de los individuos de la nueva generación se genera cruzando dos individuos de la generación actual (crossover).
- Un porcentaje muy bajo de los nuevos individuos es generado al azar, para escapar de mínimos locales (mutantes).

Biased Random Key Genetic Algorithm (BRKGA)

- Cada individuo se genera combinando un elemento seleccionado al azar del conjunto de elite y el otro de la conjunto no-elite.

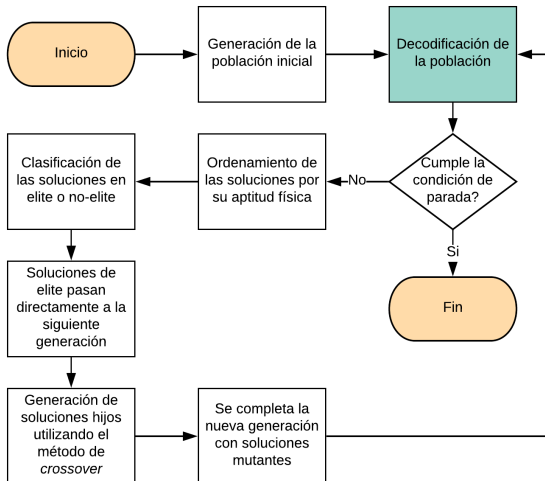
Biased Random Key Genetic Algorithm (BRKGA)

- Cada individuo se genera combinando un elemento seleccionado al azar del conjunto de elite y el otro de la conjunto no-elite.
- Parameterized Uniform Crossover. La probabilidad de que se trasmita el alelo del padre de elite es mayor que la del padre de no-elite.

Biased Random Key Genetic Algorithm (BRKGA)

- Cada individuo se genera combinando un elemento seleccionado al azar del conjunto de elite y el otro de la conjunto no-elite.
- Parameterized Uniform Crossover. La probabilidad de que se trasmita el alelo del padre de elite es mayor que la del padre de no-elite.

Diagrama de Flujo del BRKGA



Generación de la Población Inicial

- Se crea una cantidad de vectores de enteros aleatorios igual a la cantidad de soluciones por generación que se desea.

Generación de la Población Inicial

- Se crea una cantidad de vectores de enteros aleatorios igual a la cantidad de soluciones por generación que se desea.
- Los vectores tienen un tamaño igual a la cantidad de clientes de la instancia.

Generación de la Población Inicial

- Se crea una cantidad de vectores de enteros aleatorios igual a la cantidad de soluciones por generación que se desea.
- Los vectores tienen un tamaño igual a la cantidad de clientes de la instancia.
- Cada entero aleatorio del vector esta asociado a un identificador de cliente.

Generación de la Población Inicial

- Se crea una cantidad de vectores de enteros aleatorios igual a la cantidad de soluciones por generación que se desea.
- Los vectores tienen un tamaño igual a la cantidad de clientes de la instancia.
- Cada entero aleatorio del vector esta asociado a un identificador de cliente.

Ejemplo de un nuevo vector de enteros aleatorios.

Key	27	13	79	45	21	7	98	54
ClientId	1	2	3	4	5	6	7	8

Decodificación de los vectores en soluciones validas del problema

- Se ordena el vector de enteros aleatorios por el valor de la clave aleatoria de forma ascendente.

Decodificación de los vectores en soluciones validas del problema

- Se ordena el vector de enteros aleatorios por el valor de la clave aleatoria de forma ascendente.

Ejemplo del vector de enteros aleatorios ordenado.

Key	7	13	21	27	45	54	79	98
ClientId	6	2	5	1	4	8	3	7

Decodificación de los vectores en soluciones validas del problema

- Se ordena el vector de enteros aleatorios por el valor de la clave aleatoria de forma ascendente.

Ejemplo del vector de enteros aleatorios ordenado.

Key	7	13	21	27	45	54	79	98
ClientId	6	2	5	1	4	8	3	7

- Implementé dos decodificadores cada uno con su estrategia para generar soluciones.

Decodificación de los vectores en soluciones validas del problema

- Se ordena el vector de enteros aleatorios por el valor de la clave aleatoria de forma ascendente.

Ejemplo del vector de enteros aleatorios ordenado.

Key	7	13	21	27	45	54	79	98
ClientId	6	2	5	1	4	8	3	7

- Implementé dos decodificadores cada uno con su estrategia para generar soluciones.
- Ambos decodificadores generan una solución válidas del problema a partir de un vector de enteros aleatorios ordenado.

Decodificador Simple

- Los vehículos están ordenados de forma ascendente según su identificador.

Decodificador Simple

- Los vehículos están ordenados de forma ascendente según su identificador.
- Toma el primer cliente e intenta agregarlo en la ruta del primer vehículo disponible.

Decodificador Simple

- Los vehículos están ordenados de forma ascendente según su identificador.
- Toma el primer cliente e intenta agregarlo en la ruta del primer vehículo disponible.
- Si logra insertarlo repite el proceso con el siguiente cliente para el mismo vehículo.

Decodificador Simple

- Los vehículos están ordenados de forma ascendente según su identificador.
- Toma el primer cliente e intenta agregarlo en la ruta del primer vehículo disponible.
- Si logra insertarlo repite el proceso con el siguiente cliente para el mismo vehículo.
- Si no lo logra, considera que la ruta del vehículo actual esta completa e intenta agregar el mismo cliente en el siguiente vehículo disponible.

Decodificador Simple

- Los vehículos están ordenados de forma ascendente según su identificador.
- Toma el primer cliente e intenta agregarlo en la ruta del primer vehículo disponible.
- Si logra insertarlo repite el proceso con el siguiente cliente para el mismo vehículo.
- Si no lo logra, considera que la ruta del vehículo actual esta completa e intenta agregar el mismo cliente en el siguiente vehículo disponible.
- Repite hasta completar la ruta de todos los vehículos disponibles.

Decodificador Simple

- Los vehículos están ordenados de forma ascendente según su identificador.
- Toma el primer cliente e intenta agregarlo en la ruta del primer vehículo disponible.
- Si logra insertarlo repite el proceso con el siguiente cliente para el mismo vehículo.
- Si no lo logra, considera que la ruta del vehículo actual esta completa e intenta agregar el mismo cliente en el siguiente vehículo disponible.
- Repite hasta completar la ruta de todos los vehículos disponibles.

Ejemplo de la solución generada por el decodificador simple

Key	7	13	21	27	45	54	79	89
ClientId	6	2	5	1	4	8	3	7

Vehículo 1: 6 -> 2

Vehículo 2: 5 -> 1

- Se diferencia del decodificador simple en el momento en que encuentra un cliente que no entra en la ruta del vehículo actual.

Decodificador Goloso

- Se diferencia del decodificador simple en el momento en que encuentra un cliente que no entra en la ruta del vehículo actual.
- En vez de pasar al siguiente vehículo, prueba con el siguiente cliente.

Decodificador Goloso

- Se diferencia del decodificador simple en el momento en que encuentra un cliente que no entra en la ruta del vehículo actual.
- En vez de pasar al siguiente vehículo, prueba con el siguiente cliente.
- Por lo tanto por cada vehículo prueba todos los clientes en el orden dado.

Decodificador Goloso

- Se diferencia del decodificador simple en el momento en que encuentra un cliente que no entra en la ruta del vehículo actual.
- En vez de pasar al siguiente vehículo, prueba con el siguiente cliente.
- Por lo tanto por cada vehículo prueba todos los clientes en el orden dado.
- No prueba con los clientes que ya fueron asignados a otro vehículo

Decodificador Goloso

- Se diferencia del decodificador simple en el momento en que encuentra un cliente que no entra en la ruta del vehículo actual.
- En vez de pasar al siguiente vehículo, prueba con el siguiente cliente.
- Por lo tanto por cada vehículo prueba todos los clientes en el orden dado.
- No prueba con los clientes que ya fueron asignados a otro vehículo

Ejemplo de la solución generada por el decodificador goloso

Key	7	13	21	27	45	54	79	89
ClientId	6	2	5	1	4	8	3	7

Vehículo 1: 6 -> 2 -> 8

Vehículo 2: 5 -> 1 -> 3

Comparación entre Decodificadores

- Se realizó una prueba para comparar los tiempos de ejecución y aptitud de las soluciones generadas por ambos decodificadores.

Comparación entre Decodificadores

- Se realizó una prueba para comparar los tiempos de ejecución y aptitud de las soluciones generadas por ambos decodificadores.
- Se crearon 200 vectores de enteros aleatorios y se decodificaron utilizando ambos decodificadores.

Comparación entre Decodificadores

- Se realizó una prueba para comparar los tiempos de ejecución y aptitud de las soluciones generadas por ambos decodificadores.
- Se crearon 200 vectores de enteros aleatorios y se decodificaron utilizando ambos decodificadores.

Ejemplo de la solución generada por el decodificador goloso

Key	7	13	21	27	45	54	79	89
ClientId	6	2	5	1	4	8	3	7

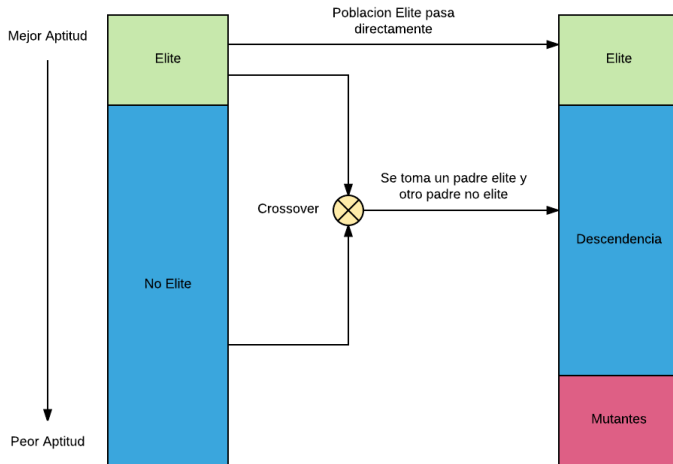
Vehículo 1: 6 -> 2 -> 8

Vehículo 2: 5 -> 1 -> 3

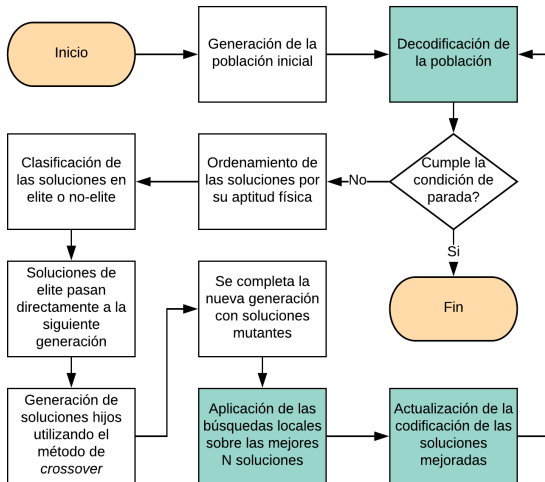
Biased Crossover



Evolución de la Población



Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local

Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local

Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local



Flow Chart del BRKGA con Búsqueda Local



Paragraphs of Text

Sed iaculis dapibus gravida. Morbi sed tortor erat, nec interdum arcu. Sed id lorem lectus. Quisque viverra augue id sem ornare non aliquam nibh tristique. Aenean in ligula nisl. Nulla sed tellus ipsum. Donec vestibulum ligula non lorem vulputate fermentum accumsan neque mollis.

Sed diam enim, sagittis nec condimentum sit amet, ullamcorper sit amet libero. Aliquam vel dui orci, a porta odio. Nullam id suscipit ipsum. Aenean lobortis commodo sem, ut commodo leo gravida vitae. Pellentesque vehicula ante iaculis arcu pretium rutrum eget sit amet purus. Integer ornare nulla quis neque ultrices lobortis. Vestibulum ultrices tincidunt libero, quis commodo erat ullamcorper id.

- Lorem ipsum dolor sit amet, consectetur adipiscing elit
- Aliquam blandit faucibus nisi, sit amet dapibus enim tempus eu
- Nulla commodo, erat quis gravida posuere, elit lacus lobortis est, quis porttitor odio mauris at libero
- Nam cursus est eget velit posuere pellentesque
- Vestibulum faucibus velit a augue condimentum quis convallis nulla gravida

Blocks of Highlighted Text

Block 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Block 2

Pellentesque sed tellus purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vestibulum quis magna at risus dictum tempor eu vitae velit.

Block 3

Suspendisse tincidunt sagittis gravida. Curabitur condimentum, enim sed venenatis rutrum, ipsum neque consectetur orci, sed blandit justo nisi ac lacus.

Heading

- 1 Statement
- 2 Explanation
- 3 Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Table

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table caption

Theorem

Theorem (Mass–energy equivalence)

$$E = mc^2$$

Example (Theorem Slide Code)

```
\begin{frame}  
\frametitle{Theorem}  
\begin{theorem}[Mass--energy equivalence]  
$E = mc^2$  
\end{theorem}  
\end{frame}
```

Figure

Uncomment the code on this slide to include your own image from the same directory as the template .TeX file.

An example of the `\cite` command to cite within the presentation:

This statement requires citation [Smith, 2012].



John Smith (2012)

Title of the publication

Journal Name 12(3), 45 – 678.

The End